International Journal of Science and Research, Vol. 1(1), 2005, pp. 21-30

Matrix Model for Web Page Community Construction

Jingyu Hou[1] and Jinli Cao[2]

[1] School of Information Technology, Deakin University, Melbourne, Vic3125, Australia
Tel: +61-3-92517448, Fax: +61-3-95217604, Email: jingyu@deakin.edu.au

[2] Department of Computer Science & CE, LaTrobe University, Melbourne, Vic3086, Australia
Tel: +61-3-94793035, Fax: +61-3-94793060, Email: j.cao@latrobe.edu.au

Abstract: The World Wide Web is now a huge information source with its own characteristics. In most cases, traditional database-based technologies are no longer suitable for web information processing and management. For effectively processing and managing web information, it is necessary to reveal *intrinsic* relationships/structures among concerned web information objects such as web pages. In this work, a set of web pages that has its own intrinsic structure is called a web page community. This paper proposes a matrix model to describe relationships among concerned web pages. Based on this model, *intrinsic* relationships among pages could be revealed, and in turn a web page community could be constructed. The issues that are related to this model in its application are deeply investigated and studied. Some applications based on this model are presented, which demonstrate the potential of this matrix model in different kinds of web page community construction and information processing.

Keywords: Web page community, matrix model, hyperlink analysis.

1. Introduction

The World Wide Web is now a huge information source. The data on the web, however, are neither raw nor very strictly typed as those in conventional database systems. This feature makes it hard to directly apply conventional techniques to process and manage information on the web. For web information processing and management, the main obstacle is the absence of a well-defined underlying data model. One approach to overcome this obstacle is to reveal intrinsic or semantic relationships/structures among concerned web data instead of defining a data model. In this work, we focus on the most commonly used information object (data) on the web - web pages (HTML documents), and define a *web page community* as a set of concerned web pages that has its own *intrinsic* structure.

The key to constructing a web page community is the intrinsic relationships among web pages. In other words, a simple gathering of web pages could not be considered as a community if there is no intrinsic relationship among them. Intrinsic relationship/structure has different meanings for different situations. For example, a set of pages that can be clustered into clusters forms a web page community; a set of pages that are relevant to a given page also forms a web page community; Kleinberg (Kleinberg 1999) considered two sets of hub pages and authority pages as two communities respectively. In order to uncover *intrinsic* relationship/structure among web pages, it is necessary to firstly model web pages and their raw relationships. The traditional approach is using vector model, i.e. each page is modeled as a keyword vector. The intrinsic relationship of pages such as page similarity is revealed by performing operations on vectors. Document object model (DOM 1998), for example, is another model for web pages that are written using markup languages such as HTML and XML. These models focus on modeling individual web page. Relationships among pages are not directly modeled.

In this paper, we propose a matrix model for web pages and community construction. Although a matrix model is widely used and it seems a straightforward approach, when the matrix model is applied within web environment, there are a lot of special issues to be addressed and resolved because of characteristics of web date. With this model, web pages, as well as their relationships, are modeled within a matrix framework. Since each page

corresponds to a row/column of the matrix, traditional vector-based techniques could also be used to reveal intrinsic relationships. In other words, the traditional vector model is a special case of this matrix model. Most importantly, intrinsic relationship among web pages could be uncovered via mathematical operations on the matrix rather than on individual vectors, which lays corresponding algorithms on a solid mathematical base. All concerned pages could be considered as a whole in terms of a matrix, and their relationships, such as similarity, correlation and cluster, could be revealed by matrix operations such as matrix decomposition, partitioning, eigenvalue and eigenvector calculation etc. Therefore, this model could be used not only for web page community construction, but also for other kinds of information processing.

This paper is organized as follow. In the next section, we propose the matrix model for web pages and community construction. In section 3, 4, 5 and 6, we discuss the issues that are related to this model, covering data space construction, noise and malicious hyperlink issue, hyperlink transitivity and decline rate, and matrix-based shortest hyperlink path algorithms. The discussion mainly focuses on hyperlink based web page community. The reasons behind it can be found in (Kleinberg 1999, Hou and Zhang 2003a). In section 7, we present a web application case study that is based on this model. Section 8 gives other examples of this model application. We conclude this work in section 9.

2. Matrix Model

Usually, a community is constructed from a set of concerned information objects, such as web pages and web access logs. For general purposes, we define a *data space* as a set of concerned information objects. Given a data space, how to model it depends on what information is used to express relationships between objects within the space. For example, given a data space that consists of a set of documents, the relationship between documents can be expressed by keywords, i.e. if two documents contain many common keywords, the relevance or similarity between these two documents is high, and vice versa. In this case, documents in this data space are modeled as keyword vectors. Considering these factors, the matrix model is a framework with the following requirements to be met:

(1) A data (information) space is constructed. For example, in a conventional database system, the data space might be the whole documents within it. But in the context of web, the situation will be complex. For different web applications, different data spaces have to be constructed.

(2) Two sets of information entities (objects), denoted as E_1 and E_2, within the constructed data space are identified. One set should be a reference system to another. That means the relationships between entities in E_1 are determined by those in set E_2, and vice versa. For example, E_1 could be a set of documents; E_2 could be a set of keywords.

The relationship between E_1 and E_2 can be classified into three classes:

(i) $E_1 = E_2$. Two sets are the same.
(ii) E_1 ? E_2. Two sets are different and in different category.
For example, E_1={documents}, E_2={keywords}.
(iii) $E_1 \sim E_2$. Two sets are different but in the same category. For example, E_1 could be one set of web pages, and E_2 could be another set of web pages.

(3) Original correlation expression between entities that belong to different sets E_1 and E_2 is defined and modeled into a matrix. The correlation expression is defined as

$$(E_1 \triangleright\triangleleft E_2)) \leftarrow CI,$$

where *CI* stands for correlation information which is the information used to describe the relationship between entities in E_1 and E_2. This expression means the correlations between entities in E_1 and E_2 are expressed by defined correlation information *CI*.

From this expression, each entity of E_1 is modeled as a row (column) of a matrix, and each entity of E_2 is modeled as a column (row) of the matrix. The values of matrix elements (intersections of rows and columns) represent the original correlation degrees between entities that belong to E_1 and E_2 separately. These original correlations degrees are determined by *CI*. For example, suppose E_1={documents}, E_2={keywords}, we can define *CI*={keywords}. Each document in E_1 could be represented as a row of a matrix, and each keyword in E_2 could be represented as a column of the matrix. If one document contains a keyword, the corresponding matrix element value is 1, otherwise is 0. If we define *CI*={weighted keywords}, the corresponding matrix element value would be the weight of the keyword rather than 1. It is clear from this example that definition of *CI* determines what information is used to express correlations between entities in two

information entity sets, and how the original correlation degrees are determined as well.

The above requirements define the matrix model for information processing and community construction. This model paves the way of revealing intrinsic relationships among information entities through matrix and/or other related mathematic operations. However, when this model is applied to practical situations, especially the web, there are some related issues to be investigated. In this work, the discussion concentrates on web pages and their hyperlinks, i.e. E_1 and E_2 are two sets of web pages and CI={hyperlinks} in the above matrix model. The ideas and methods, however, could also be applied to other kinds of correlation expressions. The following sections investigate the issues that are related to the above model requirements. The corresponding approaches and algorithms are also proposed.

3. Data Space Construction

The first requirement of the matrix model is data space construction. For traditional database, this is not a problem because the concerned data are fixed. In the context of the web, however, the situation is quite different because the web size is very huge and it is impossible to model all pages on the web within a matrix. For this reason, data space construction is critical to the success of the matrix model. It depends on what web application requirements are or what kind of web page community to be constructed.

For discussion convenience, we adapt the following concepts: if there is a hyperlink from page P to page Q, P is called a *parent* of Q and Q is called a *child* of P; if two pages have at least one common parent page, these two pages are called *siblings*. As indicated in (Mukherjea and Hara 1997), in terms of hyperlink, the semantic information about a given page u is most likely to be given by its in-view and out-view. The in-view is a set of parent pages of u, and out-view is a set of child pages of u. In other words, parent and child pages of a given page usually share some common semantic features with this page. Therefore, the data space construction in terms of hyperlink should focus on concerned pages and their parent/child pages.

Although there are many ways of constructing hyperlink-based data space (Kleinberg 1999, Hou, and Zhang 2003a, Hou and Zhang 2002), these ways in general can be classified into two categories. The first one is the method of selecting parent/child pages; the second one is that of selecting parent-child and child-parent pages. The following describes the details of these two kinds of data space construction methods.

Parent/child page selection This data space construction method is usually composed of two steps. The first step is to choose concerned pages to form a root of the data space. Secondly, the parent/child pages of each root page are selected, together with the root pages, to form the data space. This data space also includes hyperlinks between any two pages in the data space, and is considered to be a specific directed graph whose nodes are pages and edges are hyperlinks. The illustration of this method is shown in figure 1. The root of data space is located in the middle of the figure.

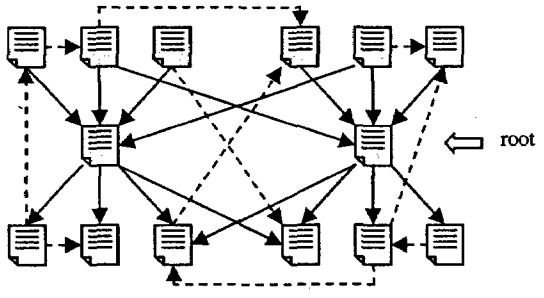

Figure 1. Data space construction from parent/child page selection

The solid line arrows represent the hyperlinks that are used to select parent/child pages of the root pages. The dashed line arrows indicate other hyperlinks that exist between pages in the data space. In practical situations, the root usually contains many concerned pages, and each root page might have many parent/child pages. It is necessary to restrict the number of parent/child pages for each root page, such that the size of the data space is reasonable (Kleinberg 1999, Hou and Zhang 2002). This kind of data space is usually used for those situations where the intrinsic relationships among the concerned pages, even among all pages in the data space, are to be uncovered such as hub/authority page finding (Kleinberg 1999), and web page clustering (Hou and Zhang 2003b).

Parent-child and child-parent page selection. This method usually consists of three steps. In the first step, the concerned pages are selected to form a root of the data space. Secondly, parent and child pages of each root page are selected. Finally, for each selected parent/child page, its child/parent

pages are selected. All selected pages in these three steps, together with their corresponding hyperlinks form the data space.

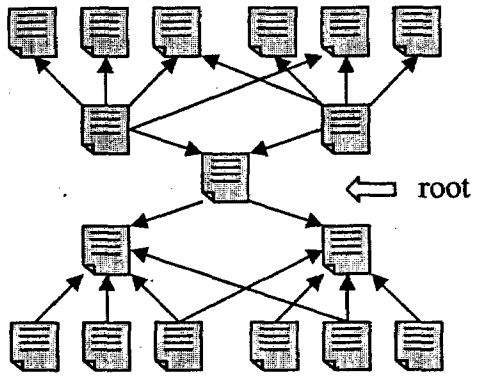

Figure 2. Data space construction from parent-child and child-parent page selection

The illustration of this method is shown in figure 2. For clearance, this figure only shows one root page. In practical situations, similar to the first method, it is also necessary to restrict the number of each page's parent/child pages in the data space. This data space is usually used for the situations where the intrinsic relationships among sibling pages and parent pages are to be uncovered such as page relevance determination. Depending on application requirements, sometimes a data space is constructed by only using parent-child or child-parent page selection instead of both at the same time.

2. Noise and Malicious Hyperlink Issue

When constructing a data space for web page communities, it is very likely that some pages that are hyperlinked have no semantic relationship. For example, the hyperlinks in the banner or index areas of a web page, as well as those pages that are pointed to by these hyperlinks, usually refer to some general information about a web site or advertisements, which are not related to the page in a semantic sense. This kind of hyperlinks/pages is called noise hyperlinks/pages. They should not be included in the data space or their influence on web page community construction should be reduced, otherwise they will distort the nature of communities.

There are two ways of eliminating or reducing the influence of noise hyperlinks/pages in a data space. The first one is to filter noise hyperlinks/pages when constructing a data space. To this end, the hyperlinks in a page are assigned semantics by the keywords around hyperlinks (i.e. anchor text) and page structure information (Chakrabarti, et al., 1998). Then the hyperlink's semantics are compared with the page semantics. If they are related (i.e. the similarity is above a certain threshold), then the hyperlink and related page are included in the data space, otherwise they are filtered. The second way is to eliminate or reduce the noise hyperlink/page influence in the process of revealing intrinsic relationships. This method is usually implemented by developing various algorithms, such as SVD based algorithm 9 (Hou and Zhang 2002) and co-citation algorithm (Garfield, 1972, Dean and Henzinger 1999, Hou and Zhang 2003b). Since hyperlinks are dynamic and there is no standard of how to identify noise hyperlinks/pages, it can be foreseen that various algorithms will be put forward and research on this issue will still be a challenge.

Malicious hyperlinks are another kind of hyperlinks that need to be addressed when constructing a data space. Malicious hyperlinks are those that are deliberately added in web pages to increase the importance of some web pages on the Web or a web site, even if these added hyperlinks have no semantic relationship with the emphasized pages. This trick will cheat web search engines and unreasonably increase the importance of some pages in the data space.

Before discussing the approaches of reducing influence of malicious hyperlinks, we firstly introduce the following concepts.

Definition 1: Two pages p_1 and p_2 are *back co-cited* if they have at least one common parent page. The number of their common parents is their *back co-citation degree*. Two pages p_1 and p_2 are *forward co-cited* if they have at least one common child page. The number of their common children is their *forward co-citation degree*.

Definition 2: The pages are *intrinsic pages* if they have same page domain name. Here the domain name is the first level of the URL string associated with a web page.

Definition 3 (Dean and Henzinger 1999)**:** Two pages are *near-duplicate pages* if (a) they each have more than 10 links and (b) they have at least 95% of their links in common.

As indicated in the above section, a data space construction usually begins with selecting a root of the data space, then growing this root to form the

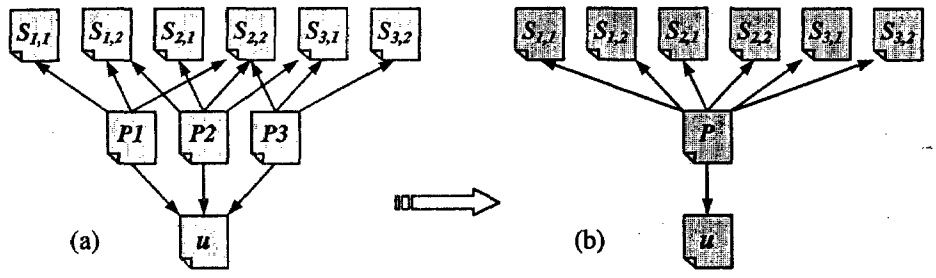

Figure 3. An example of intrinsic parent page merging

data space by adding parent/child pages of each root page. The malicious hyperlinks, therefore, are most likely to be brought into the data space by these parent/child pages. How to deal with malicious hyperlinks is now turned to be how to deal with these parent/child pages. The following is an approach of dealing with malicious hyperlinks by merging intrinsic and near-duplicate parent/child pages.

Suppose we choose a page u in the root of the data space, for pages in a web site (or server) that are hyperlinked deliberately, if some of them are imported into the data space as the parent pages of u, their children (the siblings of u) most likely come from the same site (or server), and the back co-citation degrees of these children with u would be unreasonably increased. With the merger of intrinsic parent pages, the influence of the pages from the same site (or server) is reduced to a reasonable level (i.e. the back co-citation degree of each child page with u is only 1) and the malicious hyperlinks are shielded off. For example, in figure 3, suppose the parent pages P_1, P_2, P_3 and their children $S_{1,1}$, ..., $S_{3,2}$ be intrinsic pages. In situation (a), the back co-citation degree of page $S_{2,2}$ with u is unreasonably increased to 3, which is the ideal situation the malicious hyperlink creators would like. The situation is the same for the pages $S_{1,2}$ and $S_{3,1}$. With intrinsic parent page merging, the situation (a) is changed to the situation (b) where P is a logic page representing the union of parent pages P_1, P_2, P_3, and the contribution of each child to the back co-citation degree with u is only 1, no matter how tightly these intrinsic pages are linked together.

For those sibling pages that are really relevant to the root page u and located in the same domain name as u, the intrinsic parent page merging would probably reduce their relevance to the page u. However, for data space construction, pages do not just come from a specific web site or server. Therefore the intrinsic page merging is reasonable in practical applications since one page's importance in terms of hyperlink is determined by pages in many web sites rather than a specific one. If the data space is only constructed from a specific web site or domain name, it would be unnecessary to merge intrinsic pages. From the above discussion, it is clear that there exists a trade-off between avoiding malicious hyperlinks and keeping as much semantic information as possible. The idea of this approach is the same for merging intrinsic child pages, as well as near-duplicate parent/child pages.

3. Hyperlink Transitivity and Decline Rate

Within the matrix model framework, after data space being constructed, we should select two sets of entity E_1 and E_2 in the data space and decide the correlation information CI between these two sets. In terms of hyperlink, E_1 and E_2 are two sets of web pages, and CI={hyperlinks}. Depending on application requirements, E_1 and E_2 may or may not be equal. When mapping the original correlation expression $(E_1 \triangleright\triangleleft E_2) \leftarrow CI$ into a matrix, each page in E_1 is mapped as a row (column) of the matrix, and each page in E_2 is mapped as a column (row) of the matrix. Traditionally, the matrix element value is determined as follow: if there is hyperlink from a page in E_1 to another page in E_2, then the corresponding matrix element value is set to 1, otherwise 0. This kind of correlation matrix is usually called *adjacent matrix* (Kleinberg 1999). However, the adjacent matrix only considers direct hyperlinks between any two pages in the data space. In many cases, some pages have no direct hyperlinks between them, but there is still correlation between them through other pages and hyperlinks. This hyperlink transitivity is one of the

obvious features of web data, and should be mapped into the matrix model as well. When considering hyperlink transitivity, it is worth notice that the role each page plays in the data space S is different. For instance, two kinds of pages need to be noticed. The first one is a page whose *out-link contribution* to S (i.e. the number of pages in S that are pointed to by this page) is greater than the average out-link contribution of all the pages in S.

Another kind is a page whose *in-link contribution* to S (i.e. the number of pages in S that point to this page) is greater than the average in-link contribution of all the pages in S. The pages of the first kind are called *index* pages in (Botafogo and Shneiderman 1991) (*hub* pages in (Kleinberg 1999)), and those of the second kind are called *reference* pages in (Botafogo and Shneiderman 1991) (*authority* pages in (Kleinberg 1999)). These pages are most likely to reflect certain topics within the data space S. If two pages are linked by or linking to some pages of these kinds, these two pages are more likely to be located in the same topic group and semantically related.

It is also worth notice that index pages in common sense, such as personal bookmark pages and index pages on some special-purpose web sites, might not be the index pages in the data space S if their out-link contribution to S is below the average out-link contribution in S. For the same reason, some pages with high in-degrees on the web, such as home pages of commonly used search engines, might not be the reference pages in S. Usually, we filter the home pages of commonly used search engines (e.g. *Yahoo!*, *AltaVista*, *Google* and *Excite*) from S, since these pages are not related to any specific topics. To label the importance of each page within the data space, we define a weight for each page.

For a page P_i in the data space S, we denote its weight as w_i ($0 < w_i = 1$). Given weight for each page in S, we are able to define a weight for each hyperlink between any two pages in S. This hyperlink weight is the function of page weights that are linked by this hyperlink. In other words, suppose there are two hyperlinked pages P_i and P_j in the data space S and their page weights are w_i and w_j respectively, then their hyperlink weight is defined as $w_{i,j} = f(w_i, w_j)$, where f is a function and $0 < w_{i,j} = 1$. How to define web page and hyperlink weight is still a challenge problem. One solution to this problem can be found in (Hou and Zhang 2003b).

With page and hyperlink weight, we could map transitivity correlations between pages in the data space into a matrix. Before proposing the mapping method, we firstly give the following definitions.

Definition 4. If page A has a direct link to page B, then the *length of path* from page A to page B is 1, denoted as $l(A,B) = 1$. If page A has a link to page B via n other pages, then $l(A,B) = n+1$. The *distance* from page A to page B, denoted as $sl(A,B)$, is the shortest path length from A to B, i.e. $sl(A,B) = min(l(A,B))$. The length of path from a page to itself is zero, i.e. $l(A,A) = 0$. If there are no links (direct or indirect) from page A to page B, then $l(A,B) = 8$.

It can be inferred from this definition that $l(A,B) = 8$ does not imply $l(B,A) = 8$, because there might still exist links from page B to page A.

Definition 5. *Decline rate*, denoted as F ($0<F<1$), is a variable that measures the correlation decline rate between two page with direct link, i.e. if page A has a direct link to page B with hyperlink weight $w_{A,B}$, then the correlation degree from page A to page B is $w_{A,B}F$.

How to determine the value of decline rate F to more precisely reflect the correlation relationship between pages is beyond the scope of this work. Further research could be done in this area. Since we mainly concentrate on hyperlink transitivity mapping here, for simplicity, we suppose the value of F is a constant (e.g. ½ in (Weiss, et al., 1996)).

With above definitions, a correlation degree between any two pages can be defined. This correlation degree depends on the value of decline rate F, the distance between the two pages (the farther the distance, the less the correlation degree), and weights of involved hyperlinks along the shortest path. The following definition gives this dependency function.

Definition 6. The *correlation degree* from page i to page j, denoted as c_{ij}, is defined as

$$c_{ij} = w_{i,k1} w_{k1,k2} \cdots w_{kn,j} F^{sl(i,j)},$$

where F is the decline rate, $sl(i,j)$ is the distance from page i to page j, and $w_{i,k1}$, $w_{k1,k2}$, ..., $w_{kn,j}$ are hyperlink weights respectively between the adjacent pages i, $k1$, $k2$, ..., kn, j that form the distance $sl(i,j)$, i.e. $i \rightarrow k1 \rightarrow k2 \rightarrow \ldots \rightarrow kn \rightarrow j$. If $i = j$, then c_{ij} is defined as 1.

For two web page sets E_1 and E_2 in a data space S, we suppose the size of E_1 (i.e. the number of pages

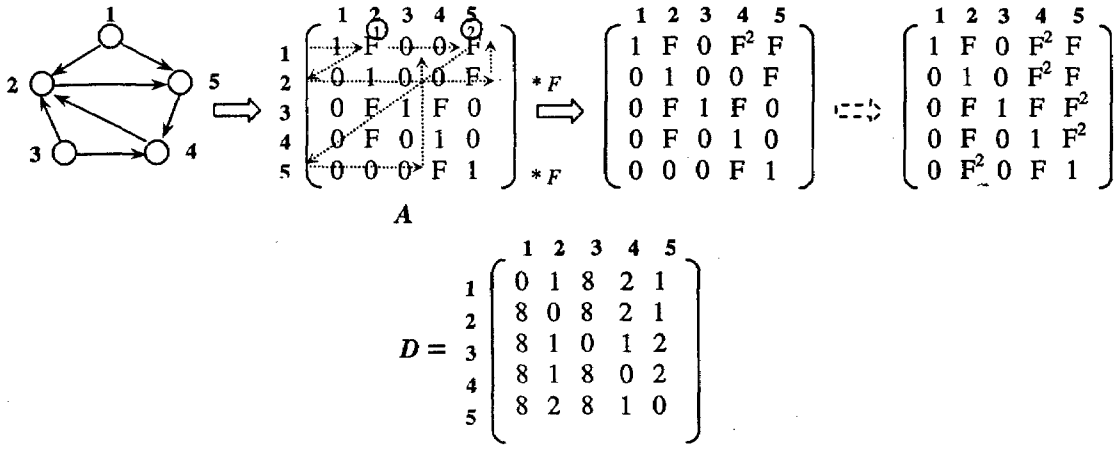

Figure 4. An example of shortest path computing algorithm

in E_1) is m, the size of E_2 is n and denote $E = E_1 \cup E_2$. Then hyperlink-based transitive correlation degrees of all the pages in E can be mapped into a $(m+n)x(m+n)$ matrix $C = (c_{ij})_{(m+n)x(m+n)}$, called *correlation matrix*. This mapping incorporates hyperlink transitivity, decline rate and page importance.

The key to computing correlation degree c_{ij} in definition 6 is the distance $sl(i,j)$ between any two pages i and j in E. The following section proposes an algorithm of computing distance $sl(i,j)$ within a matrix framework.

4. Shortest Path Finding Algorithm

The shortest path (distance) in definition 6 can be computed via some operations on elements of a special matrix called *primary correlation matrix*. The primary correlation matrix $A = (a_{ij})_{(m+n)x(m+n)}$ is constructed as follows:

$$a_{ij} = \begin{cases} F & \text{if there is a direct link from } i \text{ to } j, i \neq j \\ 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Based on this primary correlation matrix, an algorithm of computing distance $sl(i,j)$ between any two pages i and j in E is described as follows:

Step 1: For each page $E \ni i$, choose $factor = F$ and go to step 2;

Step 2: For each element a_{ij}, if $a_{ij} = factor$, then set $k = 1$ and go to step 3. If there is no element a_{ij} ($j = 1, \ldots, m+n$) such that $a_{ij} = factor$, then go back to step 1;

Step 3: If $a_{jk} \neq 0$ and $a_{jk} \neq 1$, calculate $factor * a_{jk}$;

Step 4: If $factor * a_{jk} > a_{ik}$, then replace a_{ik} with $factor * a_{jk}$, change $k = k+1$ and go back to step 3. Otherwise, change $k = k+1$ and go back to step 3;

Step 5: Change $factor = factor * F$ and go to step 2 until there are no changes to all element values a_{ij};

Step 6: Go back to step 1 until all the pages in E have been considered.

Step 7: After element values of matrix A are updated by the above steps, the distance from page i to page j is

$$sl(i, j) = [\log a_{ij} / \log F].$$

Figure 4 gives an intuitive demonstration of the above algorithm execution. In this example, five pages (numbered 1 to 5) and their linkages are firstly mapped into a primary correlation matrix A. The dashed arrows in matrix A show the first level operation sequence ($factor = F$) of the above algorithm for page 1. The procedure of other level operations for other pages is similar except for changing the values of variable *factor* according to the above algorithm. The final updated primary correlation matrix and the corresponding distance matrix D are presented in the figure as well. It is clear that although there are several paths from page 1 to page 4, the distance from page 1 to page 4 is 2, which is consistent with the real situation. The situation is the same for page 3 and page 5 in this example. This algorithm could be incorporated in correlation degree computing in definition 6.

5. Model Application Case Study

This section presents one application case study of the matrix model. The case is about web page clustering based on hyperlink analysis. The study focuses on how to meet the model requirements to guarantee the success of this model in clustering applications. All hyperlink analyses and web page clustering are conducted within a matrix framework.

In (Hou and Zhang 2003b], we proposed a matrix based clustering algorithms from hyperlinks. For a set of web pages that are to be clustered using their hyperlink information, the data space S of this algorithm is constructed by the parent/child page selection method in section 3. In this case, the second requirement of the matrix model in section 2 is satisfied by setting $E_1 = E_2 = S$. The correlation information is the correlation degree between pages that is defined in definition 6, which incorporates hyperlink transitivity and decline rate. Therefore, the pages in the data space S is modeled into a correlation matrix with the correlation expression ($S \triangleright\triangleleft S$) $\leftarrow CI$, where $CI \doteq$ {correlation degrees}.
For the convenience of discussion, we express the data space $S = R \cup V$ where R is the root set that is formed by the pages to be clustered, and V is the setof R's parent and child pages. If the number of pages in R is m, and the number of pages in V is n, the correlation matrix C then is an $(m+n)\times(m+n)$ matrix. For simplicity, C is divided into four blocks (sub-matrices) as follow:

$$C = (c_{ij})_{(m+n)\,x\,(m+n)} \begin{array}{c} \\ R \\ V \end{array} \begin{array}{c} \begin{array}{cc} R & V \end{array} \\ \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \end{array}_{(m+n)\times(m+n)}$$

In the correlation matrix C, the row vector that corresponds to each page i in R is in the form of

$$row_i = (c_{i,1}, c_{i,2}, ..., c_{i,m+n}), \quad i = 1,2,...,m.$$

From the construction of matrix C, it is known that row_i represents *out-link* relationship of page i in R with all the pages in S, and element values in this row vector indicate the correlation degrees of this page to the linked pages. Similarly, the column vector is in the form

$$col_i = (c_{1,i}, c_{2,i}, ..., c_{m+n,i}), \quad i = 1,2,...,m,$$

representing *in-link* relationship of page i in R with all the pages in S, and its element values indicate the correlation degrees from the pages in S to page i.

Each page i in R, therefore, is represented as two correlation vectors: row_i and col_i. For any two pages i and j in R, their *out-link similarity* is defined as

$$sim_{i,j}^{out} = \frac{(row_i, row_j)}{\| row_i \| \cdot \| row_j \|},$$

where

$$(row_i, row_j) = \sum_{k=1}^{m+n} c_{i,k} c_{j,k}, \quad \| row_i \| = (\sum_{k=1}^{m+n} c_{i,k}^2)^{1/2}$$

Similarly, their *in-link similarity* is defined as

$$sim_{i,j}^{in} = \frac{(col_i, col_j)}{\| col_i \| \cdot \| col_j \|}.$$

Then the similarity between any two pages i and j in R is defined as

$$sim(i, j) = \alpha_{ij} \cdot sim_{i,j}^{out} + \beta_{ij} \cdot sim_{i,j}^{in} \qquad (7.1)$$

where α_{ij} and β_{ij} are the weights for out-link and in-link similarities respectively. They are determined dynamically as:

$$\alpha_{ij} = \frac{\| row_i \| + \| row_j \|}{MOD_{ij}}, \quad \beta_{ij} = \frac{\| col_i \| + \| col_j \|}{MOD_{ij}},$$

where, $MOD_{ij} = \|row_i\| + \| row_j \| + \|col_i \| + \| col_j \|$. With the page similarity (7.1), another $m\times m$ symmetric matrix SM, called *similarity matrix* for R, can be constructed as $SM = (sm_{i,j})_{m\times m}$ for all the pages in the root set R, where

$$sm_{i,j} = \begin{cases} sim(i, j) & \text{if } i \neq j, \\ 1 & \text{if } i = j. \end{cases}$$

The matrix-based web page clustering is then implemented by partitioning the page similarity matrix SM.

With the iterative partition of the similarity matrix, hierarchical web page clusters are produced (Hou and Zhang 2003b). This clustering procedure is depicted in figure 5.

The algorithm evaluation results presented in (Hou and Zhang 2003b) are satisfactory.

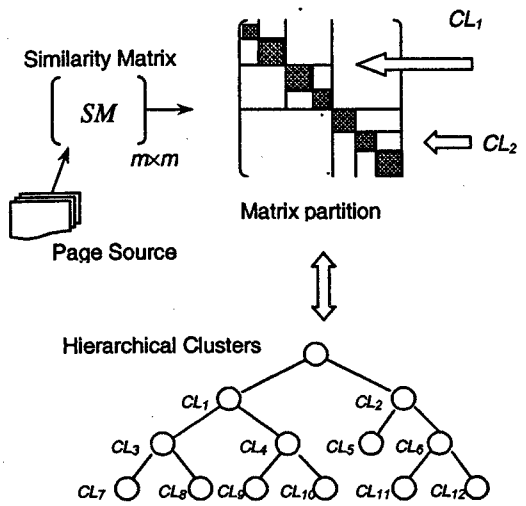

Figure 5. Matrix-based hierarchical clustering diagram

6. Other Applications

In this section, we present other applications that are also based on the matrix model in section 2. These applications demonstrate the potential of this matrix model in web community construction and other kinds of information processing.

Noise Page Elimination This problem arises from a web application that finds hub and authority pages from a data space (Kleinberg 1999). As indicated in many work (Chakrabarti, at al., 1998, Bharat and Henzinger 1998, Dean and Henzinger 1999), the data space of this application usually contains pages/hyperlinks that are not related to the concerned topics. These pages are called noise pages. If these pages are in high linkage density, they will dominate the hub/authority page finding algorithm and the obtained hub/authority might be irrelevant to the concerned topics. This phenomenon is called *topic drift* problem.

To eliminate noise pages from the data space, we (Hou and Zhang 2002) proposed a noise page elimination algorithm (NPEA) using this matrix model. The data space is the same as that of (Kleinberg 1999) which is constructed using the parent/child page selection method. Actually, the data space construction begins with a selection of root page set R, R is then grown by adding parent/child pages of R to form the final data space B. For eliminating noise pages, two matrices are built to model two correlation expressions: one for $(R \triangleright\triangleleft R) \leftarrow CI$, another one for $((B\text{-}R) \triangleright\triangleleft R) \leftarrow CI$, where CI = {hyperlinks}.

Based on these matrix models, NPEA is proposed to use singular value decomposition (SVD) of matrix to eliminate noise factors in R and B-R, and use this purified R as a reference system to eliminate noise pages from B-R. The experimental evaluation of this algorithm shows the effectiveness of this algorithm.

Relevant Page Finding. This application problem is described as follow (Hou and Zhang 2003a, Dean and Henzinger 1999): given a web page u, find a set of pages that are semantically related to it. The critical issue of this application is how to construct a data space for this given page such that the data space is rich in semantic related pages and is of reasonable size. In (Hou and Zhang 2003a), the data space is constructed from a special root set which only contains this given page u. Then the parent/child and child/parent page selection method is used to construct the required data space. This construction also incorporates techniques of dealing with malicious hyperlinks. Within this data space, C = {child pages of u}, P = {parent pages of u}, FS = {parent pages of C} and BS = {child pages of P}. The extended co-citation algorithm in (Hou and Zhang 2003a) finds relevant pages directly from FS and BS. Another algorithm, latent linkage information (LLI) algorithm, of (Hou and Zhang 2003a) is based on matrix models. Two matrices are built to model two correlation expressions: one for $(FS \triangleright\triangleleft C) \leftarrow CI$, another is for $(BS \triangleright\triangleleft P) \leftarrow CI$, where CI = {hyperlinks}. Relevant pages are found by LLI algorithm which takes advantage of SVD of these two matrices. It was found in the experiments that extended co-citation algorithm and LLI algorithm could find more semantic web pages.

Non-Web Applications. One of the representatives of this kind of applications is matrix based textual information retrieval (Berry, et al., 1995, Deerwester, et al., 1990), which finds semantic related documents from their keywords even if these documents do not share the same keywords. The corresponding method is called Latent Semantic Indexing (LSI). In LSI, E_1={documents}, E_2={keywords} and CI={weighted keywords}. A matrix is constructed to model this correlation expression $(E_1 \triangleright\triangleleft E_2) \leftarrow CI$. SVD is then applied to this matrix to reveal important associative relationships between keywords and documents that are not evident in individual documents. As a consequence, an intelligent indexing for textual information is implemented. (Papadimitriou, et al.,

1997) studied the LSI method using probabilistic approaches and indicated that LSI in certain settings is able to uncover semantically "meaningful" associations among documents with similar patterns of keyword usage, even when they do not actually use the same keywords.

9. Conclusions

Matrix model in this work could be widely used in various kinds of information processing, especially in web page community construction. To guarantee the effectiveness and success of this matrix model, the data space should be carefully constructed, and the correlation information for representing the relationship between data items in the data space must be identified. In terms of web page hyperlink analysis, data space construction depends on web application requirements, and correlation information should consider hyperlink transitivity and transitivity decline rate in some cases. Many successful applications demonstrate the effectiveness of this matrix model in web page community construction and other kinds of information processing. The related aspects of this model are also challenge research areas within which many problems need to be solved in the future.

Reference

Berry, M. W.; Dumais, and O'Brien, G. W. (1995), Using Linear Algebra S. T. for Intelligent Information Retrieval. *SIAM Review*, Vol. 37, No. 4, pp.573-595, 1995.

Bharat, K. and Henzinger, M. (1998), Improved Algorithms for Topic Distillation in a Hyperlinked Environment, *Proc. the 21st International ACM Conference of Research and Development in Information Retrieval (SIGIR98)*, pp 104-111, 1998.

Botafogo, R. A. and Shneiderman, B. (1991), Identifying Aggregates in Hypertext Structures, *Proceedings of Hypertext'91*, pp 63-74, December 1991.

Chakrabarti, S.; Dom, B., Gibson, D.; Kleinberg, J.; Raghavan, P. and Rajagopalan S. (1998), Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text, *Proc. the 7th International World Wide Web Conference*, pp 65-74, 1998.

Dean, J. and Henzinger, M. (1999), Finding Related Pages in the World Wide Web, *Proc. the 8th International World Wide Web Conference*, pp 389-401, 1999.

Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer T. K. and Harshman R. (1990), Indexing by Latent Semantic Analysis. *J. Amer. Soc. Info. Sci.*, 41(6), pp.391-407, 1990.

Document Object Model (DOM 1998) Level 1 Specification Version 1.0. *http://www.w3.org/TR/REC-DOM-Level-1*, 1998.

Garfield, E. (1972), Citation Analysis as a Tool in Journal Evaluation, *Science*, pp 471-479, 178(1972).

Hou, J. and Zhang, Y. (2003a), Effectively Finding Relevant Web Pages from Linkage Information, *IEEE Transactions on Knowledge & Data Engineering (TKDE)*, pp 940-951, Vol. 15, No. 4, July/August 2003.

Hou, J. and Zhang, Y. (2003b), Utilizing Hyperlink Transitivity to Improve Web Page Clustering, *Proceedings of the 14th Australasian Database Conference (ADC2003)*, February 4-7, 2003, Adelaide, Australia.

Hou, J. and Zhang, Y. (2002), Constructing Good Quality Web Page Communities, *Proceedings of the 13th Australasian Database Conferences (ADC2002)*, pp 65-74, Melbourne, Australia, 28 January – 1 February, 2002.

Kleinberg, J. (1999), Authoritative Sources in a Hyperlinked Environment, *Journal of the ACM* 46(1999).

Mukherjea, S. and Hara, Y. (1997), Focus+Context Views of World-Wide Web Nodes, *Proceedings of the 8th ACM Conference on Hypertext (Hypertext97)*, pp 187-196, 1997.

Papadimitriou, C.; Raghavan, P.; Tamaki, H. and Vempala S.(1997), Latent Semantic Indexing: A Probabilistic Analysis, *Proceedings of ACM Symposium on Principles of Database Systems*, 1997.

Weiss, R.; Vélez, B.; Sheldon, M.A.; Namprempre, C.; Szilagyi, P.; Duda, A. and Gifford, D.K. (1996), HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering, *Proceedings of the Seventh ACM Conference on Hypertext*, pp 180-193, 1996.