INFORMATICA, 2007, Vol. 18, No. 2, 217–238

A Matrix-Based Model for Web Page Community Construction and More

Jingyu HOU
School of Engineering and Information Technology, Deakin University
Melbourne, Vic 3125, Australia
e-mail: jingyu@deakin.edu.au

Received: May 2005

Abstract. The rapid development of network technologies has made the web 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 have their intrinsic relationships is called a web page community. This paper proposes a matrix-based 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 the application of the model are deeply investigated and studied. The concepts of community and intrinsic relationships, as well as the proposed matrix-based model, are then extended to other application areas such as biological data processing. Some application cases of the model in a broad range of areas are presented, demonstrating the potentials of this matrix-based model.

Key words: web page community, matrix model, hyperlink analysis, bioinformatics.

1. Introduction

With more and more information being put on the web, how to manage web data is becoming more and more important and urgent. 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 data on the web. For web data 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 data on the web – web pages (HTML documents), and define a *web page community* as a set of concerned web pages that have their own *intrinsic* relationships.

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 are no intrinsic relationships among them. An intrinsic relationship has different meanings for different situations. For example, web pages that are clustered

into clusters form a web page community; web pages that are relevant to a given page also form a web page community; Kleinberg (1999) considered two sets of hub pages and authority pages as two communities respectively. In order to uncover *intrinsic* relationships among web pages, it is necessary to firstly model web pages and their raw relationships. The traditional approach is using a vector model, i.e., each page is modeled as a keyword vector. The intrinsic relationships of pages such as page similarity are 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 work, we propose a matrix-based model for web page community construction, and extend the concepts and the model that are for web page communities to other areas such as biological data processing. Matrix models are widely used in many areas. For example, El-Qawasmeh *et al.* (2004) proposed a matrix multiplication algorithm and confirmed the feasibility of using clustered workstations to provide fast and low cost solutions to many computationally intensive applications. However, when matrix models are applied within a web environment, there are some special issues to be addressed and resolved because of the characteristics of web data. With our proposed 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 our model. Most importantly, intrinsic relationships 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. Actually, all concerned pages could be considered as a whole within a matrix framework, and their relationships, such as similarities, correlations and clusters, 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 various applications such as biological data processing.

This paper is organized as follows. In the next section, we propose a matrix-based model for web pages and community construction. In Sections 3, 4, 5 and 6, we discuss the issues that are related to the application of 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 relationships among web pages and corresponding community construction. In Section 7, we present a web application case study that is based on this model. In Section 8, we extend the concepts introduced for web page communities, as well as the proposed model, to another application area – biological data processing. We also exam how the proposed model could be used to produce meaningful results for biologists. Section 9 gives other application examples of the model. Finally, we conclude this work and indicate some further research directions in Section 10.

2. A Matrix-Based Model

A community is usually constructed from a set of concerned objects, such as web pages and web access logs. For general purposes, we define a *data space* as a set of concerned 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 more common keywords, the relevance or similarity between these two documents is higher, and vice versa. In this case, documents in this data space are modeled as keyword vectors. These vectors could form rows of a matrix. In general, a matrix-based model is a framework with the following required elements:

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

(2) Two sets of objects, denoted as E_1 and E_2, within the identified data space are formed. 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.
There are three types of relationship between E_1 and E_2:
   (i) $E_1 = E_2$. Two sets are the same in terms of size and category.
   (ii) $E_1 \neq E_2$. Two sets are different in terms of size and category. For example, $E_1 = \{\text{documents}\}$, $E_2 = \{\text{keywords}\}$.
   (iii) $E_1 \sim E_2$. Two sets are different in size 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 correlations between objects that belong to different sets E_1 and E_2 are defined and modeled into a matrix. The correlations are defined as the following expression

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

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

When we construct a matrix-based model from the above correlation expression, each object of E_1 is modeled as a row (column) of a matrix, and each object of E_2 is modeled as a column (row) of the matrix. The values of matrix entries (intersections of rows and columns) are original correlation degrees between objects that belong to E_1 and E_2 separately. These original correlations degrees are determined by CI. For example, suppose $E_1 = \{\text{documents}\}$, $E_2 = \{\text{keywords}\}$, we can define $CI = \{\text{keywords}\}$. 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 entry value is 1, otherwise is 0. If we define $CI = \{\text{weighted keywords}\}$, however, the corresponding matrix entry value would be the weight of the keyword rather than 1. It is clear that definition of CI determines what information is used to express correlations between objects in two object sets, and how the original correlation degrees are determined as well.

The above required elements define a matrix-based model for various applications such as community construction and data processing. 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 issues to be examined. For web page community construction, our discussion is based on web pages and their hyperlinks, i.e., E_1 and E_2 are two sets of web pages and $CI = \{\text{hyperlinks}\}$ in the above matrix-based model. The ideas and methods, however, could also be extended and applied to other kinds of correlation expressions in other applications such as biological data processing. In the following sections, we first address those issues that are related to the application of the model in web page community construction. The corresponding approaches and algorithms are also proposed. The extension of these results to biological data processing and other applications will be presented later.

3. Data Space Construction

Data space construction or identification is the first required element of the matrix-based model. 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. Therefore, data space construction is critical to the success of the model application. It depends on what the 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 of a web page u is most likely 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 web 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.

There are some methods for constructing hyperlink-based data spaces, such as our work in (Hou and Zhang, 2003a; Hou and Zhang, 2002) and Kleinberg's work (1999). In general, a hyperlink-based data space can be constructed by using one of the following two methods or the combination of these two methods. The first one is selecting parent/child pages; the second one is selecting parent-child and child-parent pages.

Parent/Child Page Selection

This data space construction method is 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. Fig. 1 shows the data space structure constructed from this method. The root of the data space is located in the middle of the figure. 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 therefore 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 consists of three steps. Firstly, 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 from these three steps, together with their corresponding hyperlinks, form the data space. Fig. 2 shows the data space structure constructed from this method. 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.

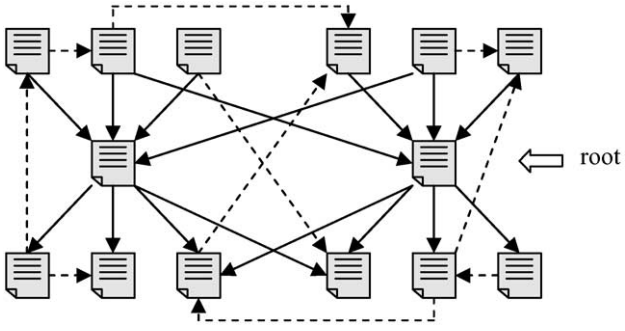

Fig. 1. Data space construction from parent/child page selection.

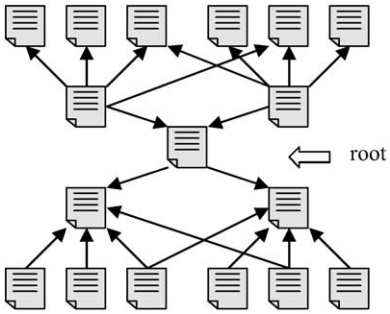

Fig. 2. Data space construction from parent-child and child-parent page selection.

4. Noise and Malicious Hyperlink Issue

When constructing a data space for web page community construction, it is very likely that some pages that are hyperlinked have no semantic relationships. For example, the hyperlinks within 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 for 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 within a page are assigned semantics by the keywords around hyperlinks (i.e., anchor text) and page structure information. 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. More details of this method can be found in (Chakrabarti *et al.*, 1998). 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. We proposed an algorithm that is based on matrix singular value decomposition (SVD) for this purpose (Hou and Zhang, 2002) and the evaluation demonstrated the effectiveness of the algorithm. There are also other algorithms of this kind such as co-citation algorithms (Garfield, 1972) in Dean and Henzinger's work (1999), as well as our work (Hou and Zhang, 2003a). Since hyperlinks are dynamic and there are no standards of what are 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 into web pages to increase the importance rate of some web pages on the Web, even if these added hyperlinks have no semantic relationships with the emphasized pages. This trick

would 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. Two or more 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 stated in the above section, a data space construction usually begins with selecting a root for the data space, then growing this root to form the 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 into 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 Fig. 3, suppose the parent pages P_1, P_2, P_3 and their children $S_{1,1}, \ldots, S_{3,2}$ be intrinsic pages. For the 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

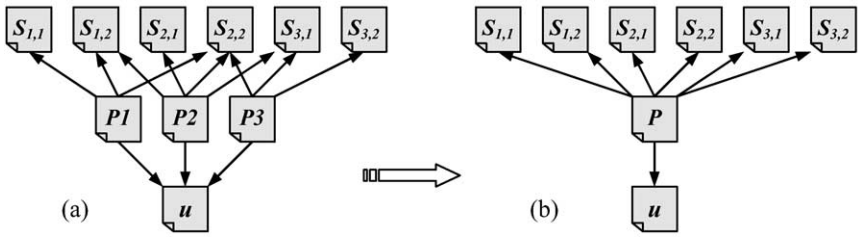

Fig. 3. An example of intrinsic parent page merging.

hyperlink creators would like. The case is the same for the pages $S_{1,2}$ and $S_{3,1}$. With intrinsic parent page merging, the situation (a) is turned into 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. The same idea could also be applied to dealing with those parent pages that come from mirrored sites.

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. This approach could also be applied to merging intrinsic child pages, as well as near-duplicate parent/child pages.

5. Hyperlink Transitivity and Decline Rate

With the matrix-based model, when mapping the original correlation expression $(E_1 \triangleright \triangleleft E_2) \leftarrow CI$ into a matrix where E_1 and E_2 are two sets of web pages and $CI = \{\text{hyperlinks}\}$, 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 entry values are determined as follow: if there is a hyperlink from a page in E_1 to another page in E_2, then the corresponding matrix entry 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-based 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. A page of the first kind is the one 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. A page of the second kind is the one 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) or *hub* pages in (Kleinberg, 1999), and those of the second kind are called *reference* pages in (Botafogo and Shneiderman, 1991) or *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. In other words, the roles that index and reference pages play are different from those of other pages.

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 $w_i (0 < w_i \leqslant 1)$. Given a 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 $0 < w_{i,j} \leqslant 1$. Since we only concentrate on hyperlink transitivity here, how to define a weight for a web page and define the function f for hyperlink weights is beyond our scope. Interested readers can read our work in (Hou and Zhang, 2003b) where we proposed weight definitions for web pages and hyperlinks.

With page and hyperlink weights, we could map transitivity correlations between pages 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 $l(A, B) = \infty$.

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

DEFINITION 5. *Decline rate*, denoted as $F(0 < F \leqslant 1)$, 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., $\frac{1}{2}$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

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

where F is the decline rate, $sl(i,j)$ is the distance from page i to page j, and $w_{i,k1}, w_{k1,k2}, \dots, w_{kn,j}$ are hyperlink weights respectively between the adjacent pages $i, k1, k2, \dots, kn, j$ that form the distance $sl(i,j)$, i.e., $i \to k1 \to k2 \to \dots \to kn \to j$. 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 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 $(m+n) \times (m+n)$ matrix $C = (c_{ij})_{(m+n)\times(m+n)}$, 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. An algorithm for computing distance $sl(i,j)$ within a matrix framework is proposed in the following section.

6. Shortest Path Finding Algorithm

The shortest path (distance) in Definition 6 can be found and calculated via some operations on the elements of a special matrix called *primary correlation matrix*. The primary correlation matrix $A = (a_{ij})_{(m+n)\times(m+n)}$ is constructed as follow

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

where $D \in (0,1)$ is a constant.

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

Step 1. For each page $i \in E$, choose $factor = D$ 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} $(j = 1, \dots, m+n)$ such that $a_{ij} = factor$, then go back to Step 1;

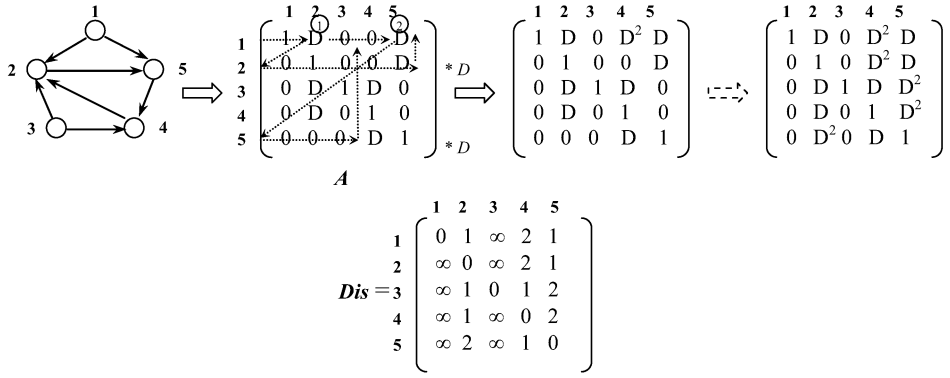

Fig. 4. An example of shortest path computing algorithm.

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

Fig. 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 ($factor = D$) 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 *Dis* 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 our observation. The situation is the same for page 3 and page 5 in this example. This algorithm could be incorporated into correlation degree computing in Definition 6.

7. A Case Study

In this section, we present an application case study of the matrix-based model. The case is about web page clustering from hyperlink analysis. The study focuses on how to meet the model requirements and how this model is applied to web page clustering. All hyperlink analyses and web page clustering are conducted within a matrix framework.

For a set of web pages that are to be clustered using their hyperlink information, the data space S is constructed using the parent/child page selection method in Section 3. 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 set of parents and children of the root pages. In this case, the second required element of the matrix-based 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 $CI = \{\text{correlation degrees}\}$.

Suppose 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:

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

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 of

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

which represents *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

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

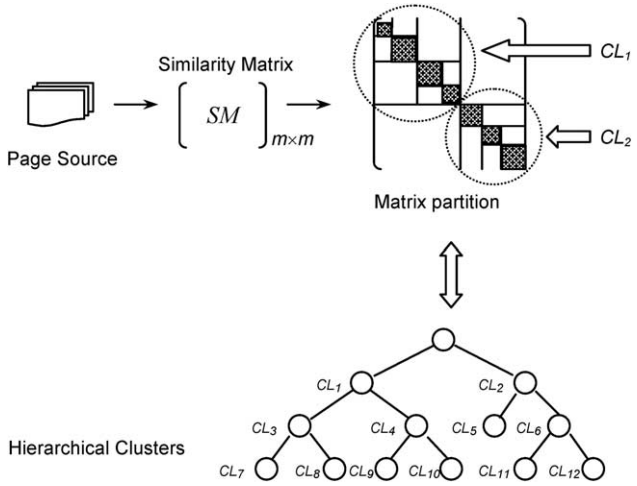

Fig. 5. Matrix-based hierarchical clustering diagram.

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

$$sim(i,j) = \alpha_{ij} \cdot sim_{i,j}^{out} + \beta_{ij} \cdot sim_{i.j}^{in}, \tag{1}$$

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 (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. Fig. 5 intuitively describes this clustering procedure. For more details of this algorithm and evaluation results, please refer to our work in (Hou and Zhang, 2003b).

8. Model Extension to Biological Data Processing

Recent development in large-scale genomic technologies, such as DNA microarray and mass spectroscope, makes it possible to obtain a lot of biological data from a single

experiment. For example, a DNA microarray chip can be used to simultaneously measure expression levels of thousands of genes in a single experiment. The advent of high-throughput data, however, also brings to researchers the challenges of how to process a large amount of biological data and reveal inherent relationships among biological objects such as genes, proteins and cells.

To cope with these challenges, the community concept proposed for web pages can be extended in terms of bioinformatics as a set of concerned biological objects that have their intrinsic relationships. For instance, genes that are clustered into clusters form a community; genes that have mutual regulatory relationships form a community, and proteins that are involved in an interaction network also form a protein community. Therefore the matrix-based model proposed in Section 2 for revealing intrinsic relationships of communities can also be directly applied to biological communities to reveal intrinsic relationships among biological objects. Accordingly, the concepts of hyperlink transitivity and decline rate, as well as related definitions, in Section 5 can be extended in the context of biological data processing, where the hyperlink transitivity is considered as the regulatory or interaction transitivity among biological objects such as genes and proteins. The related short path finding algorithm in Section 6 therefore can also be applied to biological data processing.

With the matrix-based model, it would be easy to model concerned biological data within a uniform framework and capture global data, as well as biological object, relationships when data processing is conducted. For example, a k-means method (e.g., Sultan *et al.*, 2002; Tseng and Kao, 2005) in gene clustering only takes into account local similarities between genes. The clustering results, therefore, are local optimal and sensitive to the number of genes to be clustered, as well as the choice of parameter k. In contrast, with a matrix model, it is possible to achieve global optimal clustering results through operations on the whole matrix, such as decomposition, partitioning, eigenvalue and eigenvector calculation.

To demonstrate the potentials of the matrix-based model in biological data processing, we will present some study cases. Prior to case study, we firstly introduce the singular value decomposition (SVD) of a matrix, which can be used in biological data processing.

8.1. *Singular Value Decomposition*

The singular value decomposition (SVD) of a matrix is defined as follow: let $A = [a_{ij}]_{m\times n}$ be a real $m \times n$ matrix. Without loss of generality, we suppose $m \geqslant n$ and the rank of A is $rank(A) = r$. Then there exist orthogonal matrices $U_{m\times m}$ and $V_{n\times n}$ such that

$$A = U \begin{pmatrix} \Sigma_1 \\ 0 \end{pmatrix} V^T = U\Sigma V^T, \tag{2}$$

where $U^TU = I_m, V^TV = I_n$, $\Sigma_1 = diag(\sigma_1, ..., \sigma_n)$, $\sigma_i \geqslant \sigma_{i+1} > 0$ for $1 \leqslant i \leqslant r-1$, $\sigma_j = 0$ for $j \geqslant r+1$, Σ is a $m \times n$ matrix, Σ_1 is a $n \times n$ diagonal matrix, U^T and V^T are the transpositions of matrices U and V respectively, I_m and I_n represent

$m \times m$ and $n \times n$ identity matrices separately. The *rank* of A indicates the maximal number of independent rows or columns of A. Equation (2) is called the singular value decomposition of matrix A. The singular values of A are diagonal elements of Σ (i.e., $\sigma_1, \sigma_2, ..., \sigma_n$). The columns of U are called left singular vectors and those of V are called right singular vectors (Datta, 1995; Golub and Van Loan, 1993).

The SVD can be used effectively to extract certain important properties relevant to the structure of a matrix, such as the number of independent columns or rows, eigenvalues, approximation matrix and so on (Datta, 1995; Golub and Van Loan, 1993). Since the singular values of the matrix A are in an non-increasing order, it is possible to choose a proper parameter k such that the last $r - k$ singular values are much smaller than the first k singular values, and these k singular values dominate the decomposition. The next theorem reveals this fact.

Theorem [Eckart and Young]. *Let the SVD of A be given by* (2) *and* $U = [u_1, u_2, \ldots, u_m]$, $V = [v_1, v_2, \ldots, v_n]$ *with* $0 < r = rank(A) \leqslant \min(m, n)$, *where* u_i, $1 \leqslant i \leqslant m$ *is an* m*-vector,* v_j, $1 \leqslant j \leqslant n$ *is an* n*-vector and*

$$\sigma_1 \geqslant \sigma_2 \geqslant ... \geqslant \sigma_r > \sigma_{r+1} = ... = \sigma_n = 0.$$

Let $k \leqslant r$ *and define*

$$A_k = \sum_{i=1}^{k} u_i \cdot \sigma_i \cdot v_i^T. \tag{3}$$

Then

1. *rank(*A_k*)* $= k$.
2. $\min\limits_{rank(B)=k} ||A - B||_F^2 = ||A - A_k||_F^2 = \sigma_{k+1}^2 + ... + \sigma_r^2$.
3. $\min\limits_{rank(B)=k} ||A - B||_2 = ||A - A_k||_2 = \sigma_{k+1}$.

where $||A||_F^2 = \sum_{j=1}^{n} \sum_{i=1}^{m} |a_{ij}|^2$ *and* $||A||_2^2 = \max$*(eigenvalues of* $A^T A$*) are measurements of matrix* A.

The proof of the above theorem can be found in (Datta, 1995). This theorem indicates that matrix A_k, which is constructed from partial singular values and vectors, is the best approximation matrix to A (i.e., Conclusions 2 and 3 of the Theorem) with rank k (Conclusion 1 of the Theorem). In other words, A_k captures the main information of A and trivial information in A is filtered. This important property has potentials in many application areas, such as filtering noise data from a data set and, in turn, reducing data set size in data processing.

It is recognized by many researchers that biological experiments (e.g., microarray experiments) usually produce a lot of noise data, and in most cases the amount of biological data is huge (Yang *et al.*, 2006). With the above property, SVD could deal with these noise data related problems provided the original correlations among the concerned objects (such as genes and conditions) could be modeled into a matrix. On the other hand,

since $k \leqslant r$ and matrix A_k only contains partial matrix elements of the original matrix A, the computation cost of an algorithm that is based on A_k could be reduced.

8.2. *Case 1: Eliminating Noise Information in Microarray Data Processing*

In a microarray experiment, some genes are expected to be differentially modulated in tissues under different conditions, with their expression levels increased or decreased to signify the experimental conditions (Yang *et al.*, 2006). These discriminatory genes are very useful in clinical applications and should be selected and classified from the data set. On the other hand, however, a number of genes in a microarray experiment are house-keeping genes and are unrelated to the classification task (Xiong *et al.*, 2001). Further more, microarray experiments might produce a lot of noise data due to various factors beyond control. These noise data and unrelated genes could distort those related genes in data analysis and cause the analysis results meaningless. Eliminating noise information is therefore necessary in biological data processing.

With the proposed matrix-based model, noise genes could be eliminated by making use of the merit of SVD of a matrix. Actually, in the matrix-based model, we define two sets of information entities E_1 and E_2 as:

$$E_1 = \{\text{genes}\}, \quad E_2 = \{\text{experiment conditions}\}.$$

Accordingly, we define the correlation information for E_1 and E_2 as $CI = \{$expression level values$\}$. We assume the sizes of E_1 and E_2 are m and n respectively. Therefore the correlation relationship $(E_1 \triangleright\triangleleft E_2) \leftarrow CI$ is mapped into a matrix $A = (a_{ij})_{m\times n}$, where each row represents a gene, each column represents a condition, and a_{ij} is the expression level value of gene i under the condition j. Since A is a real matrix, there is a SVD of A such that

$$A_{m\times n} = U_{m\times m}\Sigma_{m\times n}V^T_{n\times n}.$$

Suppose the rank of matrix A is $\mathit{rank}(A) = r$, the singular values in the matrix $\Sigma_{m\times n}$ are as follows:

$$\sigma_1 \geqslant \sigma_2 \geqslant ... \geqslant \sigma_r.$$

For a given threshold $\delta(0 < \delta \leqslant 1)$, we choose a parameter k such that

$$(\sigma_k - \sigma_{k+1})/\sigma_k \geqslant \delta,$$

and denote

$$U_k = [u_1, u_2, ..., u_k]_{m\times k},\ V_k = [v_1, v_2, ..., v_k]_{n\times k},\ \Sigma_k = diag(\sigma_1, \sigma_2, ..., \sigma_k),$$

where $u_i = (u_{1i}, u_{2i}, \ldots, u_{mi})^T,\ v_i = (v_{1i}, v_{2i}, \ldots, v_{mi})^T$.

Let

$$A_k = U_k \Sigma_k V_k^T.$$

As indicated in the above theorem, A_k is the best approximation matrix to A with rank k. Therefore we use the matrix A_k to take the place of A for further data processing. From the properties of SVD, it can be seen that replacing A with A_k filters noise information in the original microarray data, and the matrix A_k more reasonably reflects real situations in gene expressions.

Based on the matrix A_k, a number of algorithms could be proposed to further process microarray data. For example, the work of gene selection in (Yang *et al.*, 2006) could be improved by proposing a new algorithm on the matrix A_k instead of the original correlation matrix A. Here, however, we propose another approach for eliminating those genes that are affected by noise data.

In terms of similarity, each gene could be mapped from the matrix A_k into a k-dimensional vector in another space, i.e., gene $i(g_i)$ is the row i of the matrix $U_k \Sigma_k$

$$g_i = (u_{i1}\sigma_1, u_{i2}\sigma_2, ..., u_{ik}\sigma_k), \quad i = 1, 2, \ldots, m.$$

We define the *centroid* of all genes as a k-dimensional vector $c = (c_1, c_2, \ldots, c_k)$ where

$$c_i = \sum_{j=1}^{m} u_{ji}\sigma_i / m, \quad i = 1, 2, \ldots, k.$$

The similarity between a gene g_i and the centroid is defined as

$$s_i = \frac{|(g_i, c)|}{||g_i||_2 \cdot ||c||_2}, \quad i = 1, 2, \ldots, m,$$

where

$$(g_i, c) = \sum_{j=1}^{k} u_{ij}\sigma_j c_j, \quad ||g_i||_2 = \sqrt{(g_i, g_i)}, \quad ||c||_2 = \sqrt{(c, c)}.$$

Let $\varepsilon = \sum_{i=1}^{m} s_i / m$, a gene i that satisfies the condition $s_i < \varepsilon$ is considered as a noise affected gene and is eliminated from the gene data set.

This SVD-based data processing makes it possible to eliminate noise information in the original data set (via replacing A with A_k), and eliminate those noise-affected genes (via mapping genes into another k-dimensional sub-space). This algorithm is also a case of reducing data set size by SVD-based operations.

8.3. *Case 2: Reconstructing Protein-Protein Interaction Networks*

Schwikowski *et al.* (2000) used published protein interaction data of the yeast *Saccharomyces cerevisiae* from public databases (Costanzo *et al.*, 2000; Mewes *et al.*, 2000)

to generate a yeast protein-protein interaction network. With this network, some functions of uncharacterized proteins could be predicted from the functions of characterized interaction partner proteins.

For the reliability evaluation of the generated network in predicting protein functions, it was reported that for 1,393 characterized proteins, whose functions are known from the database, only 72% of them were correctly predicted in terms of the correctness defined in (Schwikowski *et al.*, 2000). There are more than one-quarter of characterized proteins could not be correctly predicted. One reason behind it is that there are unknown biological connections that could not be revealed from this directly generated interaction network (Schwikowski *et al.*, 2000).

Actually, the work in (Schwikowski *et al.*, 2000) just used the direct interaction data between proteins to construct the network and predict functions for uncharacterized proteins. The interaction transitivity was not taken into account, which might cause prediction incorrect for some characterized proteins. With the matrix-based model and accompanying algorithms in the above sections, the reconstruction of protein-protein interaction network could be improved as follows.

The functions of a protein are predicted in this way (Schwikowski *et al.*, 2000): if the protein has at least one characterized interaction partner, the functions of the partner(s) are sorted by frequency. The most frequent functions (e.g., at most three) of the partners are selected as indicators of functions of the protein. In other words, the partners predict the functions of the protein.

With the matrix-based model, we define the entity sets E_1 and E_2 as $E_1 = E_2 = \{\text{proteins}\}$, while the correlation information $CI = \{\text{prediction}\}$. Suppose the size of E_1 and E_2 is m, the proteins involved in network construction are modeled into a matrix $A = (a_{ij})_{m\times m}$, where a_{ij} is the prediction (correlation) degree of protein i for protein j. Its value is determined by the contribution percentage of protein i in the prediction of functions of protein j. In terms of the regulation weight, which is equivalent to the page weight defined in Section 5, a_{ij} is also the prediction (regulation) weight of protein i for protein j. For a protein k that is directly predicted by protein j with the prediction degree a_{jk}, the indirect prediction from protein i to protein k is then defined as $f(a_{ij}, a_{jk}) = a_{ij} \times a_{jk}$. In this way, if an uncharacterized protein has characterized interaction partners, its functions could be predicted and this protein becomes a characterized one. This prediction transitivity is implemented via the function f. After all transitive predictions are determined by the shortest transitive path finding algorithm in Section 6, we can go further to perform matrix based operations to find out those proteins (predictors) that most likely predict other proteins, and those proteins that are most likely predicted by the predictors. This algorithm actually consists of the following three steps, two of which have been described above:

Step 1. Map the original direct interaction and prediction information into a matrix $A = (a_{ij})_{m\times m}$.

Step 2. Apply the transitive path finding algorithm in Section 6 to improve the prediction between proteins. For simplicity, we still denote the improved matrix which contains the transitive predictions as $A = (a_{ij})_{m\times m}$. This improved matrix is also an improved

protein-protein interaction network. In addition to improving the results in (Schwikowski *et al.*, 2000) at this step, we can go further in the following Step 3 to find more from this improved network.

Step 3. Find out predictors and most likely predicted proteins. We define an m-dimensional vector $x = (x_1, x_2, \ldots, x_m)^T$, where x_i represents the weight of protein i being a predicted protein, and another m-dimensional vector $y = (y_1, y_2, \ldots, y_m)^T$, where y_i represents the weight of protein i being a predictor. Let z be an initial vector $(1, 1, 1, \ldots, 1)^T \in \mathbf{R}^m$. We conduct the following iterations for two vectors x and y:

$$x^{(k)} = (A^T A)^{k-1} A^T z, \quad y^{(k)} = (AA^T)^k z, \quad k = 1, 2, 3, \ldots,$$

where $x^{(k)}$ and $y^{(k)}$ are the kth iteration vectors of x and y respectively, A^T is the transposition of matrix A. Kleinberg (1999) proved that the vector sequence $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots\}$ converges to the principal eigenvector of $A^T A$, and the vector sequence $\{y^{(1)}, y^{(2)}, y^{(3)}, \ldots\}$ converges to the principal eigenvector of AA^T. Thus the termination of the iteration is guaranteed.

After the above iteration reaches a steady point (i.e., values of vectors $x^{(k)}$ and $y^{(k)}$ will not change any more), we choose those proteins with the highest values in vector y as predictors, while those proteins with the highest values in vector x as the most likely predicted proteins. These two kinds of proteins would play more important roles in biological experiment design and observation.

The above algorithm takes into account the transitive interaction between proteins, as well as mutual regulation relationships among concerned proteins. A similar algorithm in (Kleinberg, 1999) was successfully applied to finding authority and hub web pages in the context of hyperlinks.

9. Other Applications

In this section, we present some other applications of the proposed matrix-based model, which demonstrate the potentials of the model in various areas.

Noise Page Elimination

This problem arises from a web application that intends to find hub and authority pages from a data space (Kleinberg, 1999). As indicated in many works (Chakrabarti *et 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 hub/authority found by the algorithm might be irrelevant to the concerned topics. This phenomenon is called *topic drift*.

To eliminate noise pages from the data space, we (Hou and Zhang, 2002) proposed a noise page elimination algorithm (NPEA) that is based on this matrix-based 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 $((B-R) \triangleright \triangleleft R) \leftarrow CI$, where $CI = \{\text{hyperlinks}\}$. Based on these matrices, NPEA algorithm uses 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 key 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 the work of 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, $C = \{\text{child pages of } u\}$, $P = \{\text{parent pages of } u\}$, $FS = \{\text{parent pages of } C\}$ and $B = \{\text{child pages of } P\}$. The extended co-citation algorithm of 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 $CI = \{\text{hyperlinks}\}$. 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 are able to 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 intends to find 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, $E_1 = \{\text{documents}\}$, $E_2 = \{\text{keywords}\}$ and $CI = \{\text{weighted keywords}\}$. 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 that are with similar patterns of keyword usage, even when they do not actually use the same keywords.

10. Conclusions

The matrix-based model proposed in this work could be widely used in various application areas, such as web page community construction and biological data processing. To guarantee the effectiveness and success of the model, a data space should be carefully constructed, and the correlation information for representing the relationship between data objects in the data space must be identified. In the context 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. For biological data processing, regulatory or interaction transitivity between biological objects such as genes and proteins should be addressed as well. Many successful applications demonstrate the effectiveness and potentials of this matrix-based model in different areas. The related aspects of this model are also challenge research areas within which many problems need to be solved in the future. Further evaluations of those proposed algorithms in biological data processing need to be conducted in the near future, although some similar algorithms have already been successfully applied to other practical situations.

References

Berry, M.W., S.T. Dumais and G.W. O'Brien (1995). Using linear algebra for intelligent information retrieval. *SIAM Review*, **37**(4), 573–595.

Bharat, K., and M. Henzinger (1998). Improved algorithms for topic distillation in a hyperlinked environment. In *Proc. the 21st International ACM Conference of Research and Development in Information Retrieval (SIGIR98)*. pp 104–111.

Botafogo, R.A., and B. Shneiderman (1991). Identifying aggregates in hypertext structures. In *Proceedings of Hypertext'91*. pp 63–74.

Chakrabarti, S., B. Dom, D. Gibson, J. Kleinberg, P. Raghavan and S. Rajagopalan (1998). Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proc. the 7th International World Wide Web Conference*. pp 65–74.

Costanzo, M.C. *et al.* (2000). The yeast proteome database (YPD) and *Caenorhabditis elegans* proteome database (WormPD): comprehensive resources for the organization and comparison of model organism protein information. *Nucleic Acids Res.*, **28**, 73–76.

Datta, B.N. (1995). *Numerical Linear Algebra and Application*. Brooks/Cole Publishing Company.

Dean, J., and M. Henzinger (1999). Finding related pages in the world wide web. In *Proc. the 8th International World Wide Web Conference*. pp. 389–401.

Deerwester, S., S.T. Dumais, G.W. Furnas, T.K. Landauer and R. Harshman (1990). Indexing by latent semantic analysis. *J. Amer. Soc. Info. Sci.*, **41**(6), 391–407.

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

El-Qawasmeh, E., A. Al-Ayyoub and N. Abu-Ghazaleh (2004). Quick matrix multiplication on clusters of workstations. *Informatica*, **15**(2), 203–218.

Garfield, E. (1972). Citation analysis as a tool in journal evaluation. *Science*, **178**, 471–479.

Golub, G.H., and C.F. Van Loan (1993). *Matrix Computations*, second edition. The Johns Hopkins University Press.

Hou, J., and Y. Zhang (2003a). Effectively finding relevant web pages from linkage information. *IEEE Transactions on Knowledge & Data Engineering* (*TKDE*), **15**(4), 940–951.

Hou, J., and Y. Zhang (2003b). Utilizing hyperlink transitivity to improve web page clustering. In *Proceedings of the 14th Australasian Database Conference (ADC2003)*. pp. 49–57.

Hou, J., and Y. Zhang (2002). Constructing good quality web page communities. In *Proceedings of the 13th Australasian Database Conferences (ADC2002)*. pp 65–74.
Kleinberg, J. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, **46**(5), 604–632.
Mewes, H.W. *et al.* (2000). MIPS: a database for genomes and protein sequences. *Nucleic Acids Res.*, **28**, 37–40.
Mukherjea, S., and Y. Hara (1997). Focus+context views of world-wide web nodes. In *Proceedings of the 8th ACM Conference on Hypertext (Hypertext97)*. pp. 187–196.
Papadimitriou, C., P. Raghavan, H. Tamaki and S. Vempala (1998). Latent semantic indexing: a probabilistic analysis. In *Proceedings of ACM Symposium on Principles of Database Systems*. pp. 159–168.
Schwikowski, B., P. Uetz and S. Fields (2000). A network of protein-protein interactions in yeast. *Nature Biotechnology*, **18**, 1257–1261.
Sultan, M., D.A. Wigle, C.A. Cumbaa, M. Maziarz, J. Glasgow, M.S. Tsao and I. Jurisica (2002). Binary tree-structured vector quantization approach to clustering and visualizing microarray data. *Bioinformatics*, **18**(1), S111–S119.
Tseng, V.S., and C.P. Kao (2005). Efficiently mining gene expression data via a novel parameterless clustering method. *IEEE/ACM x Transactions on Computational Biology and Bioinformatics*, **2**(4), 355–365.
Weiss, R., B. Vélez, M.A. Sheldon, C. Namprempre, P. Szilagyi, A. Duda and D.K. Gifford (1996). HyPursuit: a hierarchical network search engine that exploits content-link hypertext clustering. In *Proceedings of the Seventh ACM Conference on Hypertext*. pp. 180–193.
Xiong, M., X. Fang and J. Zhao (2001). Biomarker identification by feature wrappers. *Genome Research*, **11**, 1878–1887.
Yang, K., Z. Cai, J. Li and G. Lin (2006). A stable gene selection in microarray data analysis. *BMC Bioinformatics*, **7**.

J. Hou received the PhD degree in computer science from University of Southern Queensland (Australia) in 2004 and PhD degree in computational mathematics from Shanghai University (China) in 1995. Since 2003, he is a lecturer at the School of Engineering and Information Technology, Deakin University, Australia. His main research interests are web mining, bioinformatics, algorithms for data mining and information retrieval, and semi-structured data models. He has published (co-author) one book on web communities and a number of papers in his research areas.

Matricinis žiniatinklio puslapių bendruomenės sudarymo modelis ir daugiau

Jingyu HOU

Greitas tinklo technologijų vystymas padarė žiniatinklį milžinišku informacijos šaltiniu su savom charakteristikom. Daugumoje atveju duomenų bazių technologijos jau yra netinkamos žiniatinklio informacijos apdorojimui ir valdymui. Efektyviam žiniatinklio informacijos apdorojimui ir valdymui reikia atskleisti būdingus žiniatinklio informacijos objektų, tokių kaip žiniatinklio puslapiai, ryšius/struktūras. Šiame darbe žiniatinklio puslapių, turinčių būdingus ryšius, aibė vadinama žiniatinklio puslapių bendruomene. Šis straipsnis siūlo matricinį modelį ryšių tarp dominančių žiniatinklio puslapių aprašymui. Remiantis šiuo modeliu būdingi puslapių ryšiai gali būti atskleisti, ir žiniatinklio puslapių bendruomenė gali būti sudaryta. Su modelio taikymu susiję klausimai yra giliai ištirti. Bendruomenės ir būdingų ryšių sąvokos bei pasiūlytas matricinis modelis yra vėliau išplėsti į kitas taikymų sritis, tokias kaip biologinių duomenų apdorojimas. Keli modelio taikymo atvejai plačiame sričių diapazone yra pristatyti, demonstruojant šio matricinio modelio potencialą.