

Sparse Matrix Reordering Schemes for Browsing Hypertext

Michael W. Berry, Bruce Hendrickson, and Padma Raghavan

ABSTRACT. Many approaches for retrieving documents from electronic databases depend on the literal matching of words in user's query to the keywords defining database objects. Since there is great diversity in the words people use to describe the same object, literal- or lexical- based methods can often retrieve irrelevant documents. Another approach to exploit the implicit higher-order structure in the association of terms with text objects is to compute the singular value decomposition (SVD) of large sparse term by text-object matrices. Latent Semantic Indexing (LSI) is a conceptual indexing method which employs the SVD to represent terms and objects by dominant singular subspaces so that user queries can be matched in a lower-rank semantic space. This paper considers a third, intermediate approach to facilitate the immediate detection of document (or term) clusters. We demonstrate both traditional sparse matrix reordering schemes (e.g., Reverse Cuthill-McKee) and spectral-based approaches (e.g., Correspondence Analysis or Fiedler vector-based spectral bisection) that can be used to permute original term by document (hypertext) matrices to a narrow-banded form suitable for the detection of document (or term) clusters. Although this approach would not exploit the higher-order semantic structure in the database, it can be used to develop browsing tools for hypertext and on-line information at a reduced computational cost.

1. Introduction

Lexical matching methods for information retrieval can be quite inaccurate when they are used for query processing. Given the common occurrence of synonyms and polysemous words, a more desirable approach for retrieval would allow users to retrieve information from databases according to a relevant topic or meaning. Latent Semantic Indexing (LSI) [BDO95, DDF⁺90] is an example of a *vector space* information retrieval model which addresses the problems of lexical matching retrieval methods by using statistically derived conceptual indices instead of individual words for retrieval. Assuming an underlying or latent structure in word

1991 *Mathematics Subject Classification.* Primary 65K50, 68P20; Secondary 05C50, 65K15.

The first author's research was supported by the National Science Foundation under grant Nos. NSF-CDA-91-15428 and NSF-ASC-92-03004.

The second author's research was supported by the U.S. Dept. of Energy, Office of Energy Research (Mathematical Sciences program) under contract No. DE-AC04-76DP00789.

The third author's research was supported by the National Science Foundation under grant No. NSF-ASC-94-11394, and by the Advanced Research Projects Agency under grant Army-DAAL33-91-C-0047.

usage that is somewhat obscured by variability in word choice, LSI employs a truncated singular value decomposition (SVD) [GL89] to estimate the structure in word usage across documents. Retrieval is then performed using the database of singular values and vectors obtained from the truncated SVD. Empirical data suggest that these statistically derived vectors are more robust indicators of meaning than individual terms when applied to a wide variety of text collections.

Retrieval methods can be applied to edge-vertex incidence matrices [OSG92] corresponding to graphs of hypertext, i.e., text objects with links or cross-references between them [BDO95]. The link structure can be represented by the nonzero patterns of the sparse *document-by-link* incidence matrices associated with hypertext [Siz94]. Whereas LSI can use relevant information stored in links, current hypertext search implementations based on keyword or string search do not usually exploit link structure. We propose a new approach for utilizing the information associated with links by permuting the corresponding document-by-link incidence matrices to reveal document and link clusters.

The primary focus of this work is to compare a variety of sparse matrix reordering schemes (spectral and symbolic) for generating narrow-banded (or clustered) nonzero patterns from hypertext incidence matrices. Such nonzero patterns allow the immediate detection of document and link clusters, and serve as textual browsers for hypertext and other similar on-line information. The detection of additional or *implicit* hypertext links is also improved using these narrow-banded nonzero patterns so as to facilitate automatic hypertext construction. Vector space information retrieval models such as LSI, which are based on spectral decompositions (e.g. SVD), can exploit banded incidence matrices through reduced indirect addressing (band storage rather than gather-scatter access) and optimal partitioning of nonzero elements (weighted term frequencies) across processors for parallel implementations of sparse matrix-vector multiplication (used by iterative methods such as Lanczos [Ber92]).

Section 2 reviews some of the basic concepts needed to understand IR models such as LSI, and provides a sample term-by-document matrix corresponding to a small text collection. In Section 3, both symbolic and spectral approaches for reordering document-by-link incidence matrices are presented. The term-by-document matrix from the constructive LSI example in Section 2 is used to explain both spectral and nonspectral approaches in Section 3. A performance comparison of the three methods from Sections 3 using sparse hypertext-based document-by-link matrices generated from the Condensed Columbia Encyclopedia, UNIX BSD 4.3 man pages, and a subset of HyperText Markup Language (HTML) pages from the National High-performance Software Exchange (NHSE) on the World-Wide-Web (WWW) is provided in Section 4. Finally, a brief summary and discussion of future work comprise Section 5.

2. Background

The initial phase of most vector space information retrieval models such as Latent Semantic Indexing [DDF⁺90, FD92], involves the construction of a term-by-document matrix. Each element of a term-by-document matrix reflects the occurrence of a particular word in a particular document, i.e.,

$$(2.1) \quad A = [a_{ij}],$$

where a_{ij} is the number of times or frequency in which term i appears in document j . As one does not expect that each word will appear in every document, the matrix A is typically sparse with rarely any noticeable nonzero structure. As discussed in [Durn91], local and global weightings can be applied to either increase or decrease the importance of terms within or among documents so that each element may be cast as

$$(2.2) \quad a_{ij} = L(i, j) \times G(i),$$

where $L(i, j)$ is the local weighting for term i in document j , and $G(i)$ is the global weighting for term i .

2.1. Singular Value Decomposition. LSI [BDO95] exploits the factorization of the matrix A into the product of 3 matrices using the singular value decomposition (SVD). Given an $m \times n$ matrix A , where $m \geq n$ and $\text{rank}(A) = r$, the singular value decomposition of A is defined as

$$(2.3) \quad A = U \Sigma V^T$$

where $U^T U = V^T V = I_n$ and $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$, $\sigma_i > 0$ for $1 \leq i \leq r$, $\sigma_j = 0$ for $j \geq r + 1$. The first r columns of the orthogonal matrices U and V define the orthonormal eigenvectors associated with the r nonzero eigenvalues of AA^T and $A^T A$, respectively. The columns of U and V are referred to as the left and right singular vectors, respectively, and the singular values of A are the diagonal elements of Σ or the nonnegative square roots of the n eigenvalues of AA^T [GL89].

As defined by Equation (2.3), the SVD is used to represent the original relationships among terms and documents as sets of linearly-independent vectors or *factor values*. Using k factors or the k -largest singular values and corresponding singular vectors one can encode (see [BDO95]) the original term-by-document matrix as a smaller (and more reliable) collection of vectors in k -space for conceptual query processing.

2.2. Sample Term-by-Document Matrix. For purposes of comparing the reordering schemes discussed in the next section, consider the small database of Bellocore technical memoranda first presented in [DDF+90]. In Table 1, a total of nine titles of technical memoranda with five of them (c1-c5) related to human-computer interaction and four of them (m1-m4) related to graph theory. All the bold-faced words in Table 1 denote keywords which are used as referents to the titles. The parsing rule used for this sample database required that keywords appear in more than one title. Of course, alternative parsing strategies can increase or decrease the number of indexing keywords (or terms).

Corresponding to the text in Table 1 is the 12×9 term-by-document matrix shown in Table 2. The elements of this matrix are the frequencies in which a term occurs in a document or title. For example, in title c5, the fifth column of the term-by-document matrix, *response*, *time*, and *user* all occur once. For simplicity, term weighting was not used to construct this sample matrix.

3. Reordering Techniques

We now consider the use of symbolic and spectral methods to permute the term-document matrix defined in Equation (2.1). The goal of such permutations is to make the detection of document (or hypertext) clusters more immediate without having to consider high-dimensional representations such as those used in LSI. One

TABLE 1. Database of titles from Bellcore technical memoranda.
Bold-faced keywords appear in more than one title.

Label	Titles
c1	Human machine interface for Lab ABC computer applications
c2	A survey of user opinion of computer system: response time
c3	The EPS user interface management system
c4	System and human system engineering testing of EPS
c5	Relation of user-perceived response time to error measurement
m1	The generation of random, binary, unordered trees
m2	The intersection graph of paths in trees
m3	Graph minors IV: Widths of trees and well-quasi-ordering
m4	Graph minors: A survey

TABLE 2. The 12×9 term-by-document matrix corresponding to the technical memoranda titles in Table 2.

Terms	Documents								
	c1	c2	c3	c4	c5	m1	m2	m3	m4
computer	1	1	0	0	0	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
time	0	1	0	0	1	0	0	0	0
user	0	1	1	0	1	0	0	0	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0

desirable form for the detection of such clusters is a *banded* or nearly diagonal matrix in which all the nonzero values (weighted term frequencies) fall within a band in each row and column. Specifically, the nonzero values should all fall near the line from the upper left to the lower right of the matrix. Such a nonzero structure (or pattern) facilitates the identification (demonstrated in Section 4.3) of term or document clusters having similar meaning and context.

3.1. Metrics for Evaluating Term-Document Matrix Reorderings. Term-document matrices are sparse, nonsymmetric and typically, highly overdetermined. As mentioned above, it is desirable that these matrices be reordered so that nonzeros are clustered evenly about a line from the upper left to the lower right corner of the matrix. This line, though visually a diagonal, is not the conventional diagonal of a nonsquare matrix. We define metrics suitable for evaluating reorderings by adapting some well established metrics used in symmetric matrix computations.

The *bandwidth* (\mathcal{B}) and *envelope size* (\mathcal{E}) are two measures used in the context of symmetric sparse matrices reordered to a band form. Let C be an $n \times n$ symmetric matrix; β_i , the bandwidth of row i , is the difference between i (the row number) and the smallest the column number j such that $C_{i,j} \neq 0$. Let β be the maximum of bandwidth values over all rows, i.e., $\beta = \max_{i=1}^n \{\beta_i\}$. The *bandwidth* is defined as $\mathcal{B} = 2\beta - 1$. We chose this definition over other alternatives (such as defining the bandwidth as β) because its natural extension to overdetermined term-document matrices seems more suitable as a metric for evaluating reorderings. The *envelope* of C incorporates the variation in bandwidth over all rows. The envelope size \mathcal{E} is defined by

$$\mathcal{E} = \sum_{i=1}^{i=n} \beta_i.$$

Observe that these terms capture the *distance* from the diagonal to the farthest nonzero on each row of the matrix.

To extend these definitions to evaluate reorderings of a nonsymmetric $m \times n$ term-document matrix A , we consider the straight line from the upper left (row 1, column 1) to the lower right (row m , column n) corner. The equation to this visual diagonal line can be easily computed; the *diagonal* subscript d_i of a row r_i is defined as the abscissa obtained using r_i as the ordinate value in this equation. To account for nonsymmetry, we define β_i , the bandwidth of row i , as the largest difference between d_i and any column number j such that $A_{i,j} \neq 0$. The bandwidth \mathcal{B} and the envelope size \mathcal{E} are as defined as earlier but using the the new definition of β_i . Defining \mathcal{B} as twice the maximum over row bandwidths measures how evenly the nonzero clusters are centered about the *diagonal*.

We also provide values of γ , a quantity which measures the size of the nonzero band but unlike \mathcal{B} does not take into account the displacement from the *diagonal*. Let γ_i be the difference between the largest and smallest nonzero subscript in row i of the matrix. Define γ as the maximum of γ_i over all rows; now γ differs from \mathcal{B} in not being relative to the diagonal. However, the sum of γ_i over all rows is still the same as \mathcal{E} which was defined earlier in terms of β_i .

3.2. Sample Hypertext Matrices. Four hypertext matrices used for performance comparisons among the symbolic and spectral-based methods presented in Sections 3.3 through 3.5 are listed in Table 3. The first two matrices, MAN1 and MAN2, were constructed from the *See Also* entries of the BSD 4.3 Unix manpages. The manpage of `who`, for example, contains the *See Also* entries `getuid` and `utmp`. Hence, two links are associated with `who`, namely `who↔getuid` and `who↔utmp`. Parsing all 625 manpages produced 1853 links, and hence the rows and columns of MAN1 correspond to links and manpages, respectively. The nonzero elements of both MAN1 and MAN2 are all 1's and simply reflect the incidence rather than frequency of linkage. The MAN2 matrix was derived from the MAN1 matrix by removing duplicate links (i.e., `who↔getuid` is the same as `getuid↔who`) and removing 18 manpages (columns) whose links were not connected to the main graph of the MAN1 matrix. The resulting MAN2 matrix had 1426 unique links corresponding to 607 manpages.

The 550 entries under the letter A of the Concise Columbia Encyclopedia (1989 Second Edition, on-line version) produced the 1778 cross-references or links for the CCE-A matrix listed in Table 3. For the 14-th entry ABDOMEN shown below, there are five cross-references or links indicated by brackets: [stomach], [liver], [gall

bladder], [pancreas], and [kidneys]. Hence, the 14-th column of **CCE-A** has five nonzeros (or 1's) in row positions 25 ([stomach]) through 29 ([kidneys]), which correspond to the links in order of their occurrence in the text.

ABDOMEN in vertebrates, portion of the trunk between the diaphragm and lower pelvis. In humans the abdominal cavity is lined with a thin membrane, the peritoneum, which encloses the [stomach], intestines, [liver], and [gall bladder]. The [pancreas], [kidneys], urinary bladder, and, in the female, reproductive organs are also located within the abdominal cavity.

The **NHSE** matrix in Table 3 was derived from 400 of the distributed HTML documents accessible from the National HPC Software Exchange (or **NHSE**) [**BDG-95**] homepage¹ on the World Wide Web (WWW). The selected documents fall under the **NHSE**'s *HPCC Programs and Activities* heading, and a breadth-first tree search of links of the form `` to 3 third-level links which produced a total of 4233 hypertext links.

The density and average number of nonzeros per row (μ_r) and column (μ_c) for each of the four hypertext matrices are also provided in Table 3. The *Density* of each matrix is defined to be the ratio (Nonzeros) / (Rows \times Columns). Since these matrices are quite sparse with only 1 or 2 links associated with each document (manpage, article, or HTML page).

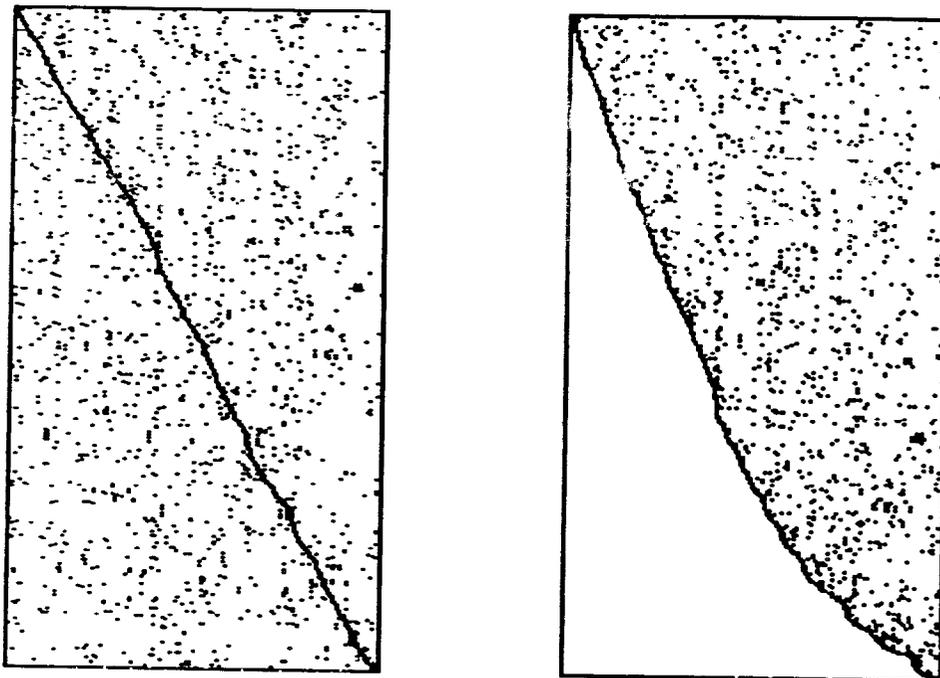
TABLE 3. Sparse hypertext matrix specifications.

Data	Source	Rows	Columns	Nonzeros	Density(%)	μ_c	μ_r
MAN1	BSD 4.3	1853	625	3706	0.003	5.9	2.0
MAN2	BSD 4.3	1426	607	2852	0.003	4.7	2.0
CCE-A	C-U Press	1778	850	2388	0.002	2.8	1.7
NHSE	Univ. TN	4233	400	5119	0.003	12.8	1.3

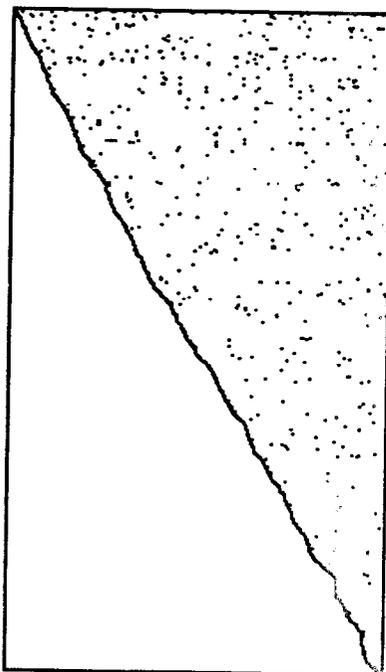
Table 4 lists the bandwidth \mathcal{B} , envelope size \mathcal{E} , and the alternative bandwidth measure \mathcal{E}' (see in Section 3.1) for each of the hypertext matrices, and Figure 1 illustrates the nonzero patterns for three of the four matrices considered. The upper-triangular structure for the nonzeros of matrices **MAN2** and **CCE-A** reflects the identification of links (cross-references) in their order of occurrence in the original (on-line) text.

The goal of the techniques in the next two sections will be to reorder rows and columns of the term-by-document matrix to reduce both \mathcal{B} and \mathcal{E} . A similar problem for symmetric matrices arises in the context of sparse Cholesky factorization. Since the Cholesky decomposition is stable under any symmetric permutation, several schemes have been proposed to reorder the rows and columns to reduce the number of generated nonzero values during the factorization. Several of the techniques derived below have their antecedents in symmetric envelope reduction algorithms. The correspondence analysis technique described in Section 3.5 is the preferred algorithm since its expense is too large for the Cholesky reordering problem.

¹NHSE home page on the World Wide Web is accessible via the URL <http://www.netlib.org/nhse/>.



(a) With duplicate links (1853×625): MAN1 (b) Without duplicate links (1426×607): MAN2



(c) 1778 cross-references by 850 articles for CCE-A

FIGURE 1. Hypertext matrices created from BSD 4.3 Unix man-pages (MAN1, MAN2) and the Letter A of the Concise Columbia Encyclopedia (CCE-A).

TABLE 4. Profiles of the hypertext matrices prior to reordering; \mathcal{E} is the envelope size, \mathcal{B} denotes the bandwidth, and γ is the alternative (non diagonal) bandwidth measure.

label	Rows	Columns	\mathcal{E}	\mathcal{B}	γ
MAN1	1853	625	308,583	1197	600
MAN2	1426	607	231,723	1137	583
CCE-A	1778	850	119,322	1667	834
NHSE	4233	400	35,491	775	392

3.3. Symbolic Reordering Methods. The envelope minimization problem for a term-by-document (or hypertext) matrix can be formulated and solved in purely symbolic terms by reordering vertices in a suitable graph representation of the matrix. The graph methods we describe in this section are based on reorderings for sparse symmetric matrices for Cholesky factorization.

Perhaps the most widely used envelope minimization method for symmetric sparse matrices is the Reverse Cuthill-McKee (RCM) method of Alan George [Geo71] which is applied to the graph of the matrix. For an $n \times n$ symmetric matrix B , the graph $G(B) = (V, E)$ is undirected with n vertices each corresponding to a row or column and edges corresponding to each nonzero, i.e. $e_{ij} \in E$ iff $B_{ij} \neq 0$. The RCM method generates a new labeling or ordering of the rows and columns of B . Observe that if $B_{uv} \neq 0$, $B_{zv} \neq 0$, row u has been labeled, but rows v and z have not, then, to minimize the bandwidth of row u , v should be numbered as soon as possible. Furthermore, to minimize the bandwidth of row z , z should also be numbered as soon as possible after u and v . In terms of $G(B)$, notice that z is adjacent to v which is in turn adjacent to u . The RCM method makes use of this observation. The main step involves a modified breadth first search (level search) from a designated starting vertex; the modification to breadth first search is that neighbors of a given vertex are explored in increasing order of degree. The RCM numbering is obtained by reversing the breadth first search numbering, i.e., if vertex u is the i -th vertex to be explored then its RCM labeling is $n - i + 1$. This reversal was shown to produce a better envelope [LS76]. The choice of the starting vertex is very significant and a *peripheral* vertex is desired. The implementation of RCM [GL81] uses an approximation to a peripheral vertex by choosing a vertex of high *eccentricity*, i.e., a vertex whose distance to some other vertex in the graph is close to the maximum distance between any two vertices in the graph.

For nonsymmetric overdetermined hypertext matrices, bipartite graphs provide a natural extension of the graph model for symmetric matrices. For the $m \times n$ hypertext matrix A , the associated undirected bipartite graph is denoted by $H(A)$ and has m row vertices and n column vertices. The row vertices are labeled $1, 2, \dots, m$ and the column vertices are labeled $1, 2, \dots, n$. The graph has an edge (\hat{r}, c) between row vertex \hat{r} and column vertex c for each $A_{rc} \neq 0$. To compute reorderings of A we apply RCM to H but we maintain two distinct numbering sequences during modified breadth first search: one for the row vertices and another for the column vertices. We obtain the final reordering by reversing each of these sequences. For example, if \hat{r} is a row (column) vertex numbered \hat{k} (l) during the search, then it is given the final number $m - \hat{k} + 1$ ($n - l + 1$). Figure 2 illustrates

the main step in RCM for the 12×9 term-by-document matrix from Section 2.2 and Table 5 shows the reordered matrix.

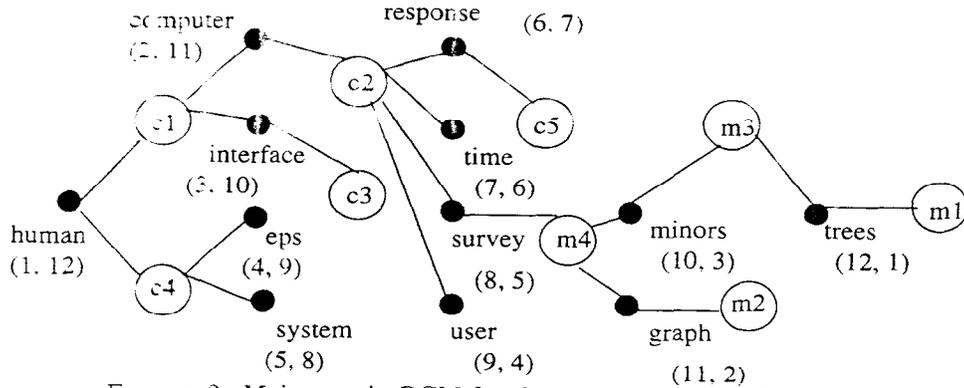


FIGURE 2. Main step in RCM for the 12×9 example; the search number, and the final RCM number are shown in parentheses for row vertices.

TABLE 5. The reordered 12×9 term-by-document matrix of the technical memoranda titles using RCM on a bipartite graph representation.

Terms	Documents								
	m1	m2	m3	m4	c5	c3	c2	c4	c1
trees	1	1	1	0	0	0	0	0	0
graph	0	1	1	1	0	0	0	0	0
minors	0	0	1	1	0	0	0	0	0
user	0	0	0	0	1	1	1	0	0
survey	0	0	0	1	0	0	1	0	0
time	0	0	0	0	1	0	1	0	0
response	0	0	0	0	1	0	1	0	0
system	0	0	0	0	0	1	1	2	0
eps	0	0	0	0	0	1	0	1	0
interface	0	0	0	0	0	1	0	0	1
computer	0	0	0	0	0	0	1	0	1
human	0	0	0	0	0	0	0	1	1

The complexity of the RCM for ordering H is proportional to the product of the maximum degree of any vertex in H and the total number of edges (nonzeros in the matrix A). For hypertext matrices with small maximum degree, the method would be extremely fast. The strength of the method is its low time complexity but it does suffer from certain drawbacks. The heuristic for finding the starting vertex is influenced by the initial numbering of vertices and so the quality of the reordering can vary slightly for the same problem for different initial numberings. Next, the overall method does not accommodate dense rows (e.g., a common link used in every document) and if a row has a significantly large number of nonzeros

it might be best to process it separately; i.e., extract the dense rows, reorder the remaining matrix and augment it by the dense rows (or common links) numbered last.

3.4. Fiedler Ordering. A recently proposed heuristic for the symmetric envelope minimization problem involves sorting the rows/columns of the matrix by the values of associated entries in the *Fiedler vector* of the graph of nonzero entries. This approach was proposed at about the same time by several different researchers [BPS93, JM92, PMGM94a, PMGM94b], and seems to often produce better orderings than more traditional combinatorial methods, albeit at a somewhat increased cost. An analysis of this approach based upon the quadratic assignment problem can be found in [GP94]. In this section the symmetric matrix technique is generalized to produce both row and column orderings for the nonsymmetric problem.

Given a graph G , with vertex set V and weighted edges E , the heuristic described in this section will use an eigenvector of L , the (weighted) *Laplacian matrix* of G . If $e_{ij} \in E$, then elements (i, j) and (j, i) of L are set equal to $-w(e_{ij})$. The diagonal is then constructed to make row sums equal to zero. More formally,

$$(3.1) \quad L(i, j) = \begin{cases} \sum_{e_{ik} \in E} w(e_{ik}) & \text{if } i = j \\ -w(e_{ij}) & \text{if } e_{ij} \in E \\ 0 & \text{otherwise.} \end{cases}$$

The Laplacian matrix has a number of nice properties. It is symmetric and positive semidefinite. The constant vector is an eigenvector with zero eigenvalue, and if the graph is connected then all other eigenvalues are positive. If the eigenvalues are sorted by increasing value, an eigenvector of L corresponding to the second eigenvalue is known as a *Fiedler vector* in recognition of the pioneering work of Miroslav Fiedler [Fie73, Fie75]. The Fiedler vector has been used in heuristics for a number of graph manipulations including partitioning [PSL90], linear labeling [JM92] and envelope minimization as alluded to above. For a survey of applications of the Fiedler vector, see [Moh91, Moh92].

The Fiedler vector has a nice interpretation which helps to explain these applications. Consider the problem of embedding a graph in the line in such a way that all the edge lengths are kept short. That is, if $e_{ij} \in E$, then vertices i and j should be near each other; particularly if the corresponding edge weight is large. Letting $x(i)$ be the location in the line of vertex i , one way to express the embedding problem mathematically is to try to minimize the following sum.

$$F(x) = \sum_{e_{ij} \in E} w(e_{ij})(x(i) - x(j))^2$$

Merely minimizing F leads to a problem with an infinite number of solutions since the minimum is invariant under translations. This can be dealt with by making the average x value equal to zero; that is, adding the constraint that $x^T e = 0$. As posed, the problem now has a trivial solution obtained by setting all the x values equal to zero. This can be avoided by adding a normalization constraint on the x vector, leading to the following problem.

$$(3.2) \quad \begin{aligned} &\text{Minimize } F(x) = \sum_{e_{ij} \in E} w(e_{ij})(x(i) - x(j))^2 \\ &\text{Subject to: } x^T e = 0 \quad \text{and} \quad x^T x = 1 \end{aligned}$$

3.5. Correspondence Analysis. Correspondence analysis [Gre84] is a geometric-based method for displaying the rows and columns of a matrix (or a two-way contingency table) as points in dual low-dimensional vector spaces. In contingency tables [Gif90] or term-by-document matrices such as the $m \times n$ matrix A defined in Equation (2.1), the cell or matrix element a_{ij} contains the frequency with which row category (keyword) i co-occurs with column category (document) j .

Define w_i as the $i \times 1$ vector of all 1's. Then, $r = Aw_n$ and $c = A^T w_m$ define vectors of row and column sums, respectively. It then follows that $\mu = w_m^T Aw_n = w_n^T c = w_m^T r$ is the sum of all the nonnegative matrix elements a_{ij} . Let D_r and D_c define diagonal matrices composed of the elements of vectors r and c , respectively. As originally described by Benzécri in [Ben73], the goal of Correspondence Analysis is to find another matrix representation (say matrix X) of the rows of A such that the Euclidean distances between rows in X approximate certain *profile*² distances between the rows of A . Simultaneously, another matrix representation (say matrix Y) of the columns of A is desired whose Euclidean distances among columns approximate certain *profile* distances between columns of A .

The squared distance or χ^2 distance δ_{ij}^2 between rows i and j of the matrix A is defined as

$$(3.3) \quad \delta_{ij}^2 = \mu \sum_k \left(\frac{a_{ik}}{r_i} - \frac{a_{jk}}{r_j} \right)^2 / c_k,$$

where r_i and c_k denote the i -th and k -th elements of the column vectors r and c , respectively. It can easily be shown that δ_{ij}^2 is the same squared Euclidean distance between rows i and j for the matrix $B = [b_{ij}]$ whose elements are defined by

$$b_{ij} = \left(\frac{a_{ij}}{r_i} \right) \left(\frac{\mu}{c_j} \right)^2.$$

The matrix B can also be written as $B = \sqrt{\mu} D_r^{-1} A D_c^{-1/2}$ so that $BB^T = XX^T$ for any Euclidean representation X of B (see [Gif90]).

One derivation of the Euclidean representation X is given by the SVD (see Equation (2.3)) of $\tilde{B} = \sqrt{\mu} D_r^{1/2} B = D_r^{-1/2} A D_c^{-1/2}$ defined by

$$(3.4) \quad \tilde{B} = \tilde{U} \tilde{\Sigma} \tilde{V}^T,$$

where $\tilde{U}^T \tilde{U} = \tilde{V}^T \tilde{V} = I_n$, and $\tilde{\Sigma} = \text{diag}(\tilde{\sigma}_1, \dots, \tilde{\sigma}_n)$. By defining

$$(3.5) \quad X = \sqrt{\mu} D_r^{-1/2} \tilde{U} \tilde{\Sigma}$$

it follows that

$$XX^T = \mu D_r^{-1/2} \tilde{U} \tilde{\Sigma}^2 \tilde{U}^T D_r^{-1/2}.$$

Using Equation (3.4), it follows that

$$XX^T = \mu D_r^{-1} A D_c^{-1} A^T D_r^{-1} = BB^T,$$

and hence the Euclidean distances among the rows of X are equal to the Euclidean distances among the rows of B .

²The term *profile* as used in [Gre84] refers to a set of relative frequencies found in the representation of a data matrix as a long, flat table having frequencies in each row expressed as percentages of their respective row sums.

The matrix X by construction will have one column of constant elements (or equal to u_n with appropriate scaling). This can be easily shown by first noting that the matrix \tilde{B} has a singular triplet corresponding to the largest singular value $\tilde{\sigma}_1 = 1$, i.e., $\{D_r^{-1/2}w_m, 1, D_c^{-1/2}w_n\}$. This triplet can be derived from the following equalities:

$$\tilde{B}D_r^{-1/2}w_n = D_r^{-1/2}AD_c^{-1/2}(D_c^{1/2}u_n) = D_r^{-1/2}Aw_n = D_r^{1/2}w_m, \text{ and}$$

$$\tilde{B}^T D_c^{-1/2}w_m = D_c^{-1/2}A^T D_r^{-1/2}(D_r^{1/2}w_m) = D_c^{-1/2}A^T w_m = D_c^{1/2}w_n.$$

Consequently, the scaled right singular vector (corresponding to $\tilde{\sigma}_1 = 1$) given by $1/\sqrt{\mu}D_r^{1/2}w_n$ has unit length so that $\sqrt{\mu}D_r^{-1/2}1/\sqrt{\mu}D_r^{1/2}w_m = w_m$ is a column of the matrix X . This constant column of X does not contribute to the distance between any two rows of X , and can be removed by computing the SVD

$$\tilde{B} - \frac{1}{\mu} \left(D_r^{1/2}w_m w_n^T D_c^{1/2} \right) = \tilde{U} \tilde{\Sigma} \tilde{V}^T,$$

which deflates or prevents the trivial singular vectors $(D_r^{-1/2}w_m, D_c^{-1/2}w_n)$ from occurring. Hence, this correction yields a Euclidean representation of

$$B - \frac{1}{\mu} \left(D_r^{1/2}w_m w_n^T D_c^{1/2} \right),$$

rather than that of the matrix B .

Whereas the rows of the matrix X (see Equation (3.5)) have the same profile distances of the rows of matrix A , the columns of the matrix $Y = \sqrt{\mu}D_c^{-1/2}\tilde{V}$ have the same profile distances of the columns of A . Note that $X = D_r^{-1}AY$. As discussed in [Gif90], this suggests that the row elements of the matrix X are, in fact, the *center of gravity* or *centroid* of the elements of the column elements of A , weighted by their frequency in the row profile³.

If the matrix representation Y for the matrix A had initially been sought, χ^2 distances between columns of A (as opposed to the rows of A) would be used to produce column elements of the matrix Y which are at the centroid of the row elements of A . Using the same SVD from Equation (3.4), the derived matrix Y and corresponding matrix X are given by

$$Y = \sqrt{\mu}D_c^{-1/2}\tilde{U}\tilde{\Sigma}, \quad X = \sqrt{\mu}D_r^{-1/2}\tilde{V}.$$

An alternative formulation for the matrix X determined by correspondence analysis is discussed in [Gre84]. Here the *generalized* singular value decomposition [GL89]

$$(3.6) \quad A = \tilde{U}\tilde{\Sigma}\tilde{V}^T, \text{ where } \tilde{U}^T D_r \tilde{U} = I_n = \tilde{V}^T D_c \tilde{V},$$

is used to minimize

$$(3.7) \quad \|A - X\|_{D_c, D_r}^2 \equiv \sum_{i=1}^m r_i (A_i - X_i)^T D_c (A_i - X_i),$$

where A_i and X_i are the i -th rows of the matrices A and X , respectively, and $D_r = \text{diag}(r_1, r_2, \dots, r_m)$. The k -largest *generalized* singular triplets from Equation (3.6) provide the optimal matrix X in Equation (3.7) of rank k [Mir66].

³Benzécri [Ben73] referred to this as *le principe barycentrique*.

For the right generalized singular vectors $\tilde{V} = [\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_n]$ from Equation (3.6), the first k column vectors $\tilde{v}_1, \dots, \tilde{v}_k$ are referred to as the k principal axes of the rows of A . The total *variation* or *inertia* of the matrix A or how well A is represented along the k principal axes is given by

$$(3.8) \quad \|A\|_{D_c, D_r}^2 = \sum_{i=1}^k r_i A_i^T D_c A_i = \sum_{i=1}^k \tilde{\sigma}_i^2,$$

where $\tilde{\sigma}_i$ is the i -th largest generalized singular value (diagonal element of $\tilde{\Sigma}$) from Equation (3.6). The *unexplained* variation when approximating A via $A_k = \tilde{U}_k \tilde{\Sigma}_k \tilde{V}_k^T$, where the subscript k denotes the first k columns of each factor of the generalized singular value decomposition in Equation (3.6), is given by

$$\|A - A_k\|_{D_c, D_r}^2 = \sum_{i=k+1}^n \tilde{\sigma}_i^2.$$

Since the total inertia is decomposed along the principal axes [Gre84], the i -th principal axis accounts for an amount of $\tilde{\sigma}_i^2$ of the total inertia in Equation (3.8).

When used as permutation vectors for the rows and columns of an $m \times n$ matrix A , the k principal axes of the rows and columns of A (i.e., $\{\tilde{v}_2, \dots, \tilde{v}_k\}$ and $\{\tilde{u}_2, \dots, \tilde{u}_k\}$, respectively) as determined by Equations (3.4) or (3.6) can produce a reordering of the original matrix A having more banded (or block diagonal) form [Gre84]. Typically, the elements of the second largest left and right generalized singular vectors ($\{\tilde{u}_2, \tilde{v}_2\}$) are sorted in ascending order to produce the required row and column permutations. Table 7 illustrates the reordering using the second largest generalized singular triplets from Equation (3.6) when A is the 12×9 matrix defined in Table 2.

TABLE 7. The reordered 12×9 term-by-document matrix of the technical memoranda titles using Correspondence Analysis.

Terms	Documents								
	c4	c1	c3	c5	c2	m4	m3	m2	m1
human	1	1	0	0	0	0	0	0	0
EPS	1	0	1	0	0	0	0	0	0
interface	0	1	1	0	0	0	0	0	0
system	2	0	1	0	1	0	0	0	0
computer	0	1	0	0	1	0	0	0	0
user	0	0	1	1	1	0	0	0	0
time	0	0	0	1	1	0	0	0	0
response	0	0	0	1	1	0	0	0	0
survey	0	0	0	0	1	1	0	0	0
minors	0	0	0	0	0	1	1	0	0
graph	0	0	0	0	0	1	1	1	0
trees	0	0	0	0	0	0	1	1	1

4. Performance on Hypertext Matrices

In this section, examples of the reduction in envelope size (\mathcal{E}) and bandwidth (\mathcal{B}) for the test collection of hypertext matrices listed in Table 3 are provided. Execution times (in elapsed CPU seconds) for the symbolic (RCM) and spectral (Fiedler and Correspondence Analysis) on a Sun SPARCstation 20 (50 MHz) are also provided. The notation CACS(i) and CANC(i) is used to represent the cases when Correspondence Analysis (see Section 3.5) is used with and without χ^2 distances (see Equation (3.3)), respectively, for the i -th largest pair of principal axes. With CANC(1), for example, $D_r = I_m$ and $D_c = I_n$ (i.e., no weighting) so that $\tilde{B} = A$ in Equation (3.4). In this case, row and column permutations are solely determined by the left and right singular vectors of A (\tilde{u}_1, \tilde{v}_1) corresponding to the largest singular value $\tilde{\sigma}_1$.

4.1. Bandwidth Reduction. As indicated by values in Tables 8 through 10, \mathcal{E} , \mathcal{B} , and γ are very large for the hypertext matrices in their natural ordering. Values in Table 8 show that \mathcal{E} is substantially reduced for all orderings with the largest envelope reduction obtained by the Fiedler vector approach discussed in Section 3.4. With respect to \mathcal{B} , however, the RCM ordering achieves the greatest bandwidth reduction (see Table 9). Notice that for some matrices (CCE-A, NHSE), the value of γ shown in Table 10 is significantly lower than that of \mathcal{B} . This is due to the clustering of nonzeros in a *narrow* band but the band itself is significantly displaced from the *diagonal*. Also, the orderings using Correspondence Analysis without χ^2 distances (see Section 3.5) tend to produce γ 's more similar to those of RCM than the Fiedler approach.

Table 11 illustrates the effects of choosing different pairs of principal axes for Correspondence Analysis with χ^2 distances (CACS). Since the 8-largest generalized singular values (see Equation 3.6) for these matrices were all approximately equal to 1 (i.e., form a cluster of generalized singular values near 1), it is not clear which pairs of principal axes best explains the variation in Equation (3.8). By selecting the 10-th pair (or 10-th largest) of principal axes, a reduction in \mathcal{E} of 43% and 17% can be obtained for MAN1 and NHSE, respectively. CACS(10) also achieves an average reduction of 25% for \mathcal{B} and 29% for γ for these matrices.

TABLE 8. Envelope size (\mathcal{E}) of the hypertext matrices after reorderings: CACS(2) denotes the use of Correspondence Analysis using χ^2 distances and the second principal axes, and CANC(1) denotes the use of Correspondence Analysis with no χ^2 distances and the first principal axes.

Matrix	Envelope Size (\mathcal{E})				
	Original	RCM	Fiedler	CACS(2)	CANC(1)
MAN1	308,583	154,240	73,554	113,413	147,399
MAN2	231,723	123,959	65,305	62,278	119,746
CCE-A	119,322	12,380	6,420	11,399	37,540
NHSE	35,491	18,937	11,058	17,725	19,821

Figures 3 through 5 illustrate the reorderings obtained for a few of the sample hypertext matrices discussed in Section 3.2. Of particular interest is the similarity of reorderings produced by the pairs (RCM, CANC(1)) and (Fiedler, CACS(2)) for the

TABLE 9. Bandwidth (\mathcal{B}) of the hypertext matrices after reorderings: CACS(2) denotes the use of Correspondence Analysis using χ^2 distances and the second principal axes, and CANC(1) denotes the use of Correspondence Analysis with no χ^2 distances and the first principal axes.

Matrix	Bandwidth (\mathcal{B})				
	Original	RCM	Fiedler	CACS(2)	CANC(1)
MAN1	1,197	267	599	723	335
MAN2	1,137	297	663	653	337
CCE-A	1,667	245	409	427	895
NHSE	775	179	197	375	201

TABLE 10. Non-diagonal bandwidth (γ) of the hypertext matrices after reorderings; CACS(2) denotes the use of Correspondence Analysis (CA) using χ^2 distances and the second principal axes, and CANC(1) denotes the use of CA with no χ^2 distances and the first principal axes.

Matrix	γ				
	Original	RCM	Fiedler	CACS(2)	CANC(1)
MAN1	600	214	382	511	218
MAN2	583	211	380	379	210
CCE-A	834	88	205	298	510
NHSE	392	122	131	219	120

TABLE 11. Effects of using different principal axes in Correspondence Analysis with χ^2 distances. CACS(2) and CACS(10) denote the use of the second and tenth principal axes, respectively.

Matrix	\mathcal{E}		\mathcal{B}		γ	
	CACS(2)	CACS(10)	CACS(2)	CACS(10)	CACS(2)	CACS(10)
MAN1	113,413	63,842	723	559	511	360
NHSE	17,725	14,719	375	277	219	157

MAN1 and MAN2 matrices. For the other two matrices (CCE-A, NHSE) whose average number of documents per link (see μ_r from Table 3) is smaller, the similarities were not as prominent. As shown in Figure 4, the *near-diagonal* clusterings obtained are quite different. In particular, CANC(1) produces a definite block-diagonal pattern of nonzeros but at the expense of the largest bandwidths (\mathcal{B} and γ) among all reorderings of the four test matrices (see Tables 9 and 10). The reduction in bandwidth achieved by CACS(10), i.e., using the 10-th largest pair of principal axes or left and right generalized singular vectors of A in Equation (3.6), is illustrated in Figure 5 for the MAN1 matrix.

The proper selection of principal axes for Correspondence Analysis with χ^2 distances when the hypertext matrix A has clustered generalized singular values ($\tilde{\sigma}_i$'s from Equation (3.8)) is problematic. Nevertheless, an iterative procedure which cycles through a subset of the largest generalized singular vectors of A as

computed by a Lanczos or block Lanczos SVD method (see [B⁺93]) is plausible. Future research in the use of principal axes for bandwidth reduction in the presence of clustered spectra is warranted.

4.2. Computational Time. The execution time (in elapsed CPU seconds) for the three ordering schemes discussed in Section 3 have been obtained on a Sun Microsystems SPARCstation 20 (50 MHz). The RCM reordering was implemented using the Fortran code from Sparspak [CGLN84]. The Fiedler reordering was produced using the Lanczos (with selective re-orthogonalization) software from Chaco 2.0 [HL94], and the reorderings using Correspondence Analysis were derived using the block Lanczos routine (b1s2) from SVDPACKC [B⁺93]. Table 12 lists the actual reordering times obtained by each method for the four hypertext matrices presented in Section 3.2.

TABLE 12. Elapsed CPU time (in seconds) on a Sun SPARCstation 20 (50 MHz) for each reordering method; number of multiplications by A and A^T for the spectral methods are in parenthesis).

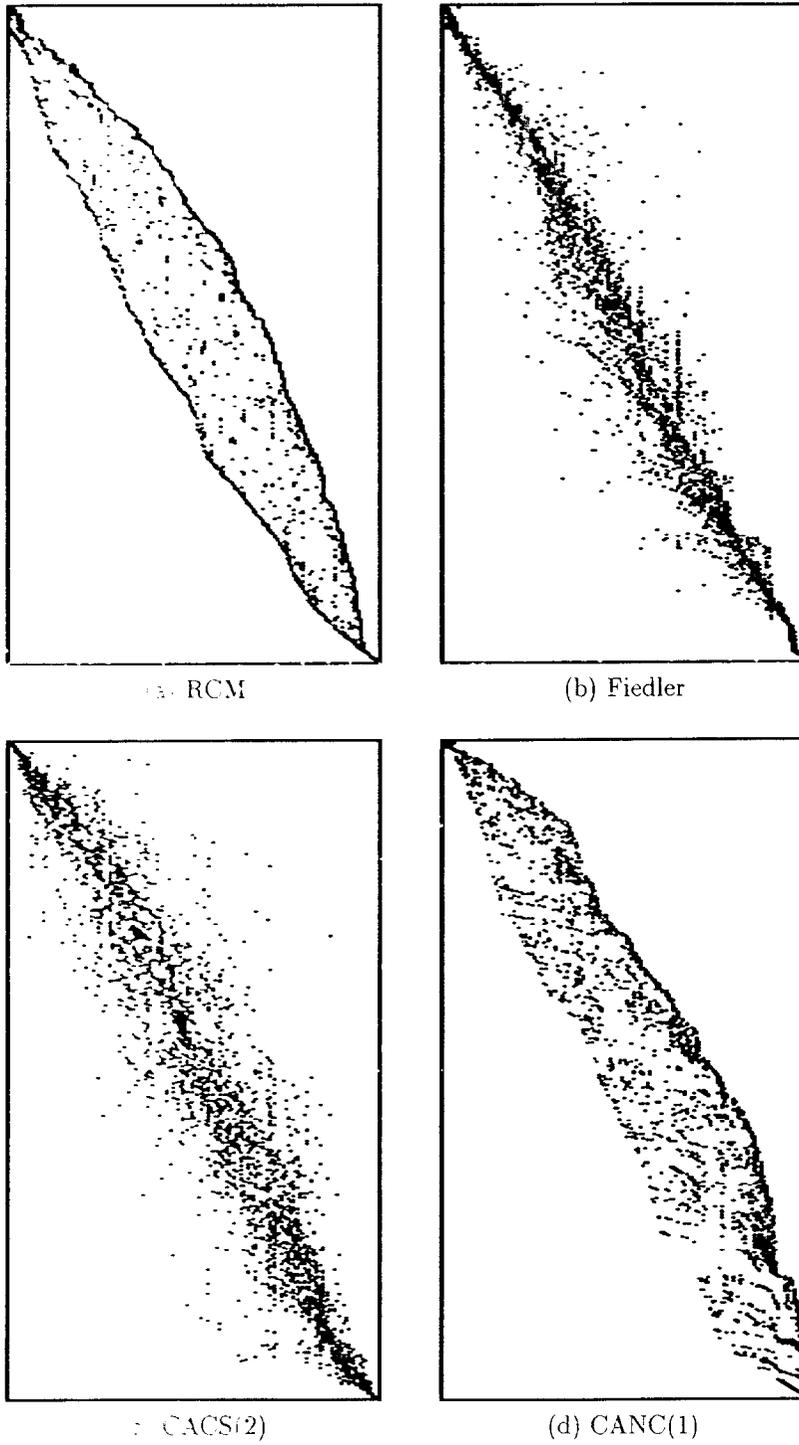
Matrix	Elapsed CPU Time (seconds)			
	RCM	Fiedler ^a	CACS(2) ^b	CANC(1) ^b
MAN1	0.019	2.36 (210)	2.00 (236)	0.96 (74)
MAN2	0.013	1.64 (180)	1.53 (199)	0.82 (74)
CCE-A	0.025	2.49 (220)	3.21 (380)	0.95 (56)
NHSE	0.037	12.72 (720)	3.18 (362)	1.43 (74)

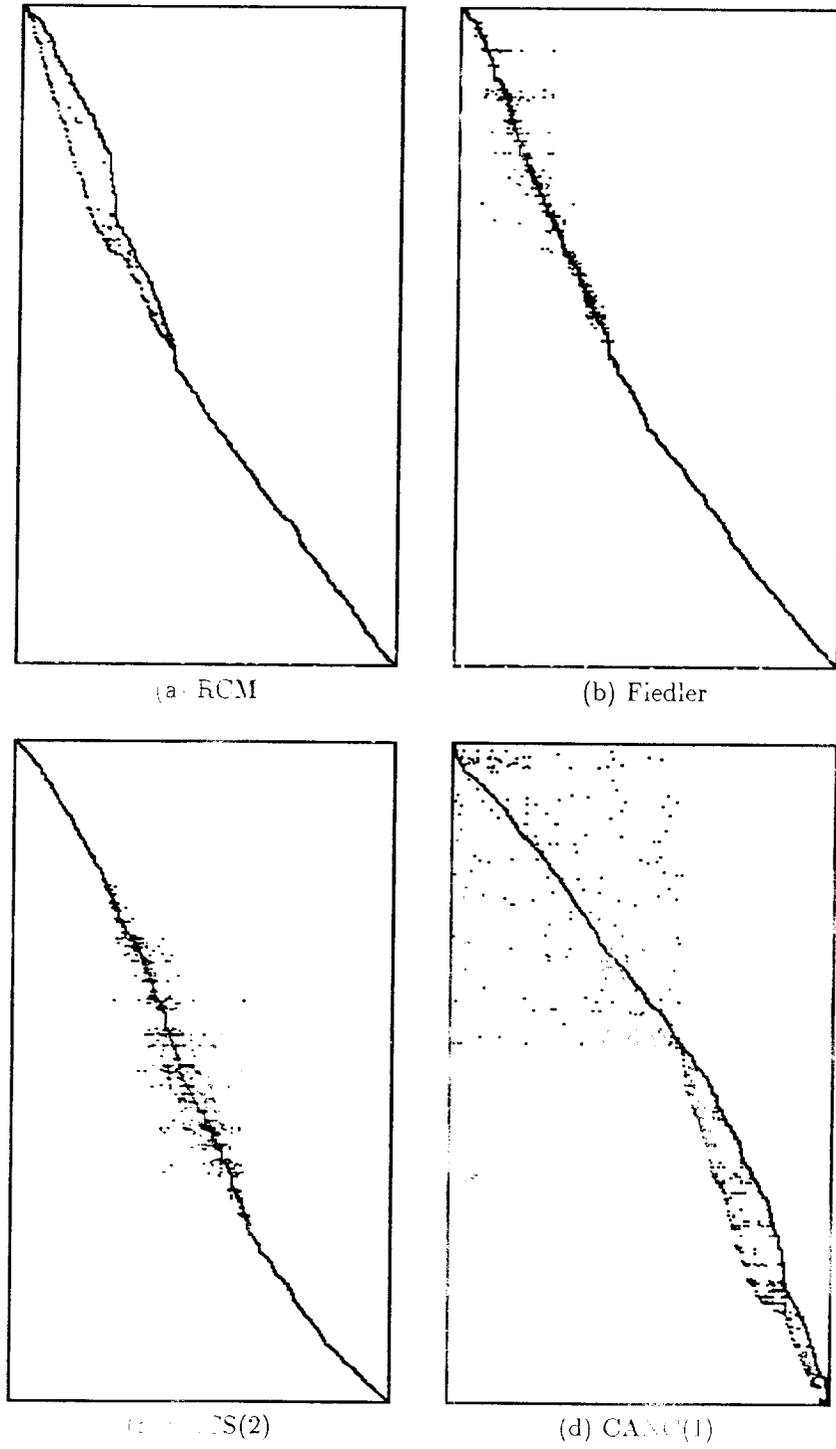
^aUsing Lanczos with selective re-orthogonalization from Chaco 2.0 [HL94].

^bUsing block Lanczos routine b1s2 from SVDPACKC [B⁺93] with a blocksize of 2 and maximum Krylov subspace dimension of 18.

Consistent with the analytic complexity discussed in Section 3.3, the execution time required by RCM is indeed low. In fact, the spectral-based reordering schemes require at least two orders of magnitude (i.e., 100 times) more execution time than RCM for these particular hypertext matrices. The dominant computational cost of the spectral-based methods involves multiplications by the sparse matrix A (and A^T) as they naturally arise in iterative methods such as Lanczos for computing singular triplets. The total number of multiplications of the form $y = Ax$ or $y^T = x^T A$ for both spectral methods is provided in Table 12. Clearly, the cost of computing the largest singular triplet (first pair of principal axes) with CANC(1) is far less than that of computing the second-largest singular triplet by CACS(2) or the second-smallest eigenvectors of the $(m+n)$ by $(m+n)$ Laplacian matrix L in Equation (3.1) by the Fiedler approach. However, as illustrated in Figures 3 and 4, the savings in sparse matrix multiplications (a factor ranging from 3 to 5 from the results in Table 12) for CANC(1) may be offset by an inferior bandwidth and envelope reduction for hypertext clustering (e.g., see Figure 4(d)). Correspondence Analysis with χ^2 distances (i.e., CACS(2)) would appear to be more competitive with respect to bandwidth (and envelope) reduction at an increased computational cost.

4.3. Browsing Clusters. The reordering of rectangular hypertext matrices can be extremely useful in the development of *visual browsers* for finding related

FIGURE 3. Reorderings of the 1853×625 MAN1 matrix.

FIGURE 7. Reorderings of the 1778×850 CCE-4 matrix.

information. Such tools can aid users in locating documents relevant to specific queries in an immediate fashion (i.e., by clusters of hypertext). From Figure 4, for example, we can extract the cluster of articles (see Figure 6) from the letter A of Condensed Columbia Encyclopedia related to people and regions of Persia around 300 BC. Notice that in graph depicted in Figure 6 there are several related articles (shown in black) not in the collection (i.e., 850 letter A articles) which are links contained in different but related letter A articles: **Demosthenes**, **Diadochi**, **Greece**, **Macedon**, **Peloponnesus**, **Persia**, and **Phillip II**. This cluster of related hypertext information is fully contained within a subwindow of each of reorderings for the 1778 links by 850 articles CCE-A matrix shown in Figure 7. The display of graphs such as that in Figure 6 coupled with windowing capabilities (e.g., mouse dragging) in a visualization tool for hypertext browsing would be highly effective for scoping the context of large and possibly distributed databases.

If the entire collection of articles (letters A through Z) of the Condensed Columbia Encyclopedia were distributed across a network (local or even the World-Wide-Web), the graph in Figure 6 as traced by the windows in Figure 7 would allow a user to selectively retrieve foreign or remote documents (e.g., articles from letters B through Z) linked to relevant local documents (e.g., articles from letter A). The relationship of remote documents with both local and other remote documents would be immediately determined by providing a *road map* of related information across the network. Without such hypertext clustering, related local documents such as **Achaea** and **Arcadia** from Figure 6 might be difficult to associate without knowing their common linkage to remote documents such as **Peloponnesus** and **Greece** a priori. That is, there would be no need to retrieve the actual texts of **Achaea** and **Arcadia** (or their links) to discover their similarity.

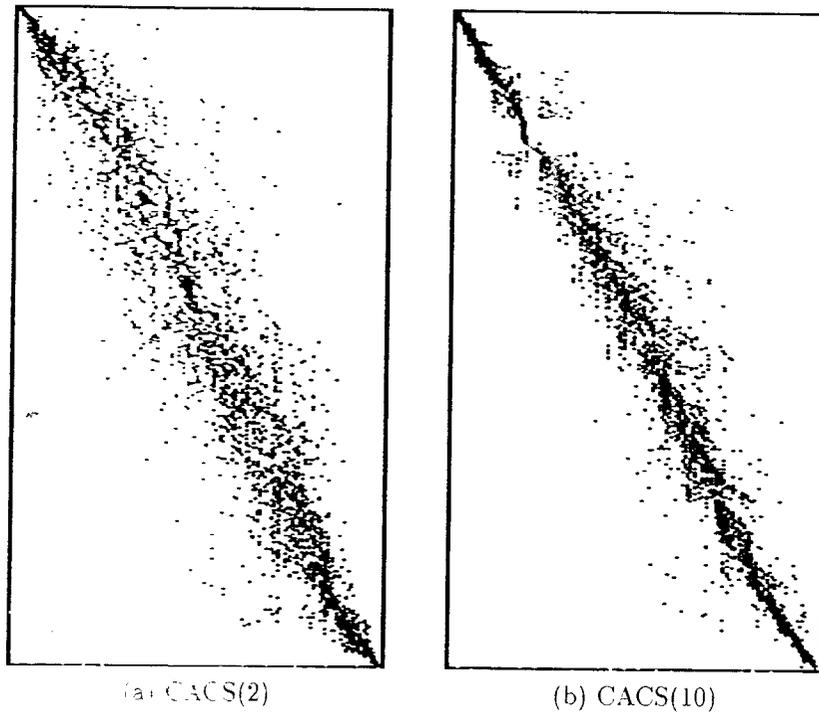


FIGURE 5. Reorderings of the 1853×625 MAN1 matrix via Correspondence Analysis with χ^2 distances for the (a) second and (b) tenth principal axes.

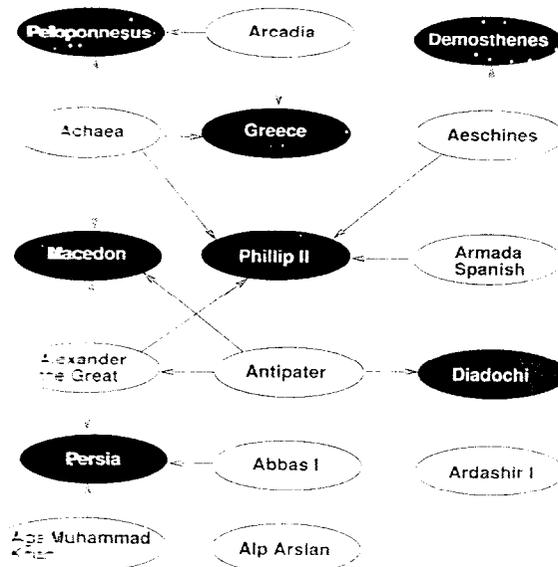


FIGURE 6. Graph of *Persia*-related articles from CCE-A for browsing.

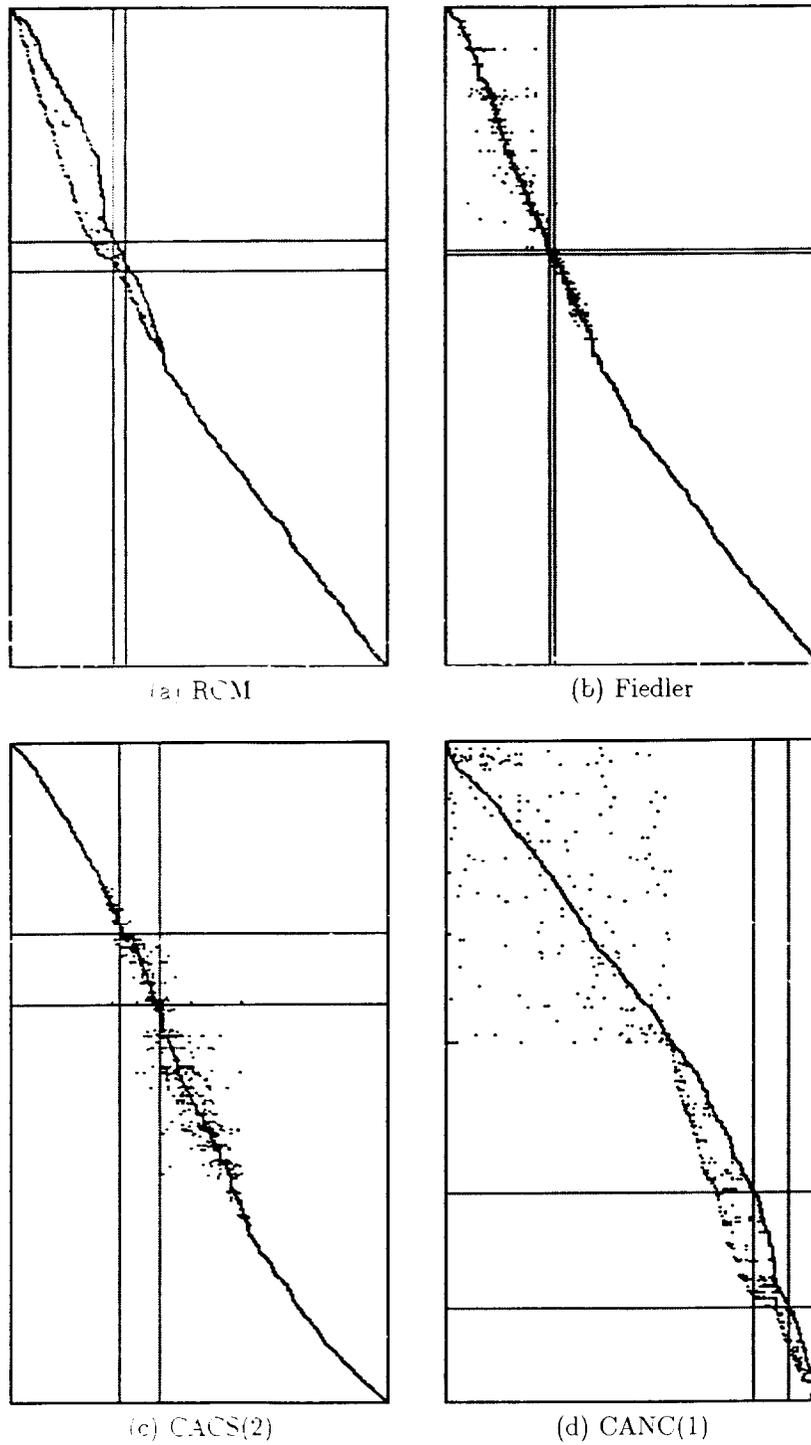


FIGURE 7. Partitioned windows containing graph of *Persia*-related CCE-A articles.

5. Summary and Future Work

Three reordering schemes have been used to produce narrow-banded hypertext matrices for cluster identification. Whereas the spectral-based methods (Fiedler and Correspondence Analysis) tend to produce matrices with smaller envelopes, the symbolic Reverse Cuthill-McKee (RCM) method produces smaller bandwidths at a substantially reduced computational cost.

The reordered hypertext matrices facilitate the development of browsing tools for isolating document clusters of related information. Such tools are greatly needed to navigate large and/or distributed databases with hypertext links. The ability to identify (without necessarily retrieving) remote documents through their links to available (local) documents on a network is possible. In addition to browsing, indexing schemes based on term-document (or hypertext) matrices such as Latent Semantic Indexing (LSI) can exploit the reorderings presented for a more equitable distribution of nonzeros across processors or nodes of a network.

Future work on the reordering of hypertext matrices involves (i) the consideration of much larger hypertext collections, (ii) the development of a visual browser tool for the X11 Release 5 Windows environment for extracting hypertext clusters of related information, (iii) the use of separator-based reorderings schemes (e.g., nested dissection), and (iv) the exploration of direct methods (as opposed to iterative schemes such as Lanczos) for computing the singular value decomposition (SVD) of banded hypertext matrices.

Acknowledgements

The authors would like to thank James Allan at the University of Massachusetts, Amherst and the anonymous referees for their helpful comments and suggestions which certainly improved the presentation and focus of this work.

References

- [E⁺93] M. W. Berry et al., *SVDPACKC: Version 1.0 User's Guide*, Tech. Report CS-93-194, University of Tennessee, Knoxville, TN, October 1993.
- [BDG⁺95] S. Browne, J. Dongarra, S. Green, K. Moore, T. Rowan, R. Wade, G. Fox, K. Hawick, K. Kennedy, J. Pool, R. Stevens, B. Olson, and T. Disz, *The National HPCC Software Exchange*, IEEE Computational Science and Engineering 2 (1995), no. 2, 62-69.
- [BDO95] M. W. Berry, S. T. Dumais, and G.W. O'Brien, *Using linear algebra for intelligent information retrieval*, SIAM Review (1995), In press.
- [Ben73] J. P. Benferhat, Tome (Vol.) 1: La Taxinomie, Tome 2: L'Analyse des Correspondances, Dunod, Paris, 1973.
- [Ber92] M. W. Berry, *Large scale singular value computations*, International Journal of Supercomputing Applications 6 (1992), no. 1, 13-49.
- [BPS93] Stephen T. Barnard, Alex Pothan, and Horst D. Simon, *A spectral algorithm for envelope reduction of sparse matrices*, Proc. Supercomputing '93, IEEE, 1993, pp. 493-502.
- [CGLN84] E. Chu, A. George, J. Liu, and B. Ng, *Sparspak: Waterloo sparse matrix package user's guide for Sparspak-A*, Tech. Report CS-84-36, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 1984.
- [DDF⁺90] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, *Indexing by direct semantic analysis*, Journal of the American Society for Information Science 41 (1990), no. 5, 391-407.
- [Dum91] S. T. Dumais, *Improving the retrieval of information from external sources*, Behavior Research Methods, Instruments, & Computers 23 (1991), no. 2, 229-236.

- [FD92] P. W. Foltz and S. T. Dumais, *Personalized information delivery: An analysis of information filtering methods*, Communications of the ACM **35** (1992), no. 12, 51–60.
- [Fie73] Miroslav Fiedler, *Algebraic connectivity of graphs*, Czech. Math. Journal **23** (1973), 298–305.
- [Fie75] Miroslav Fiedler, *A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory*, Czech. Math. Journal **25** (1975), 619–633.
- [Geo71] A. George, *Computer implementation of the finite-element method*, Tech. Report CS-208, Department of Computer Science, Stanford University, 1971.
- [Gif90] A. Gifi, *Nonlinear multivariate analysis*, John Wiley & Sons, Chichester, England, 1990.
- [GL81] A. George and J. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
- [GL89] G. Golub and C. Van Loan, *Matrix computations*, second ed., Johns-Hopkins, Baltimore, 1989.
- [GM95] Stephen Gartery and Gary L. Miller, *On the performance of the spectral graph partitioning methods*, Proc. Sixth Annual ACM-SIAM Symp. on Discrete Algs., ACM-SIAM, 1995, pp. 233–42.
- [GP94] Alan George and Alex Pothen, *An analysis of spectral envelope-reduction via quadratic assignment problems*, Tech. Report ICASE Technical Report 94-81, NASA LaRC, Hampton, VA, September 1994.
- [Gre84] M. J. Greenacre, *Theory and applications of correspondence analysis*, Academic Press, London, 1984.
- [HL94] B. Hendrickson and R. Leland, *The Chaco User's Guide, Version 2.0*, Tech. Report SAND-94-2692, Sandia National Laboratories, Albuquerque, NM, October 1994.
- [JM92] Martin Juvan and Bojan Mohar, *Optimal linear labelings and eigenvalues of graphs*, Disc. Appl. Math. **36** (1992), 153–168.
- [LS76] J. Liu and A. Sherman, *Comparative analysis of the Cuthill-McKee and Reverse Cuthill-McKee ordering algorithms for sparse matrices*, SIAM Journal of Numerical Analysis **13** (1976), 198–213.
- [Mir80] L. Mirsky, *Symmetric gage functions and unitarily invariant norms*, Q. J. Math **11** (1960), no. 1, 53–59.
- [Moh91] B. Mohar, *The Laplacian spectrum of graphs*, Graph Theory, Combinatorics and Applications, New York (Y. Alavi et al., ed.), J. Wiley, New York, 1991, pp. 871–895.
- [Moh92] B. Mohar, *Linear eigenvalues of graphs – a survey*, Disc. Math. **109** (1992), 171–183.
- [OSG92] K. S. O'Yang, B. Srinivasan, and L. M. Goldschalger, *Browsing hypertext in vector space*, Proceedings of Second International Computer Science Conference (Hong Kong), 1992, pp. 16–22.
- [PMGM94a] Glauco Paulina, Ivan Menezes, Marcelo Gattass, and Subrata Mukherjee, *Node and element resequencing using the Laplacian of a finite element graph: Part I – general concepts and algorithm*, Intl. J. Num. Methods Eng. **37** (1994), 1511–1530.
- [PMGM94b] Glauco Paulina, Ivan Menezes, Marcelo Gattass, and Subrata Mukherjee, *Node and element resequencing using the Laplacian of a finite element graph: Part II – implementation and numerical results*, Intl. J. Num. Methods Eng. **37** (1994), 1531–1555.
- [PSL90] A. Pothen, F. Simon, and K. Liou, *Partitioning sparse matrices with eigenvectors of graphs*, SIAM J. Matrix Anal. **11** (1990), no. 3, 430–452.
- [Siz94] N. L. Sizemore, *Knowledge base graph recovery using sparse matrix techniques*, Expert Systems With Applications **7** (1994), no. 2, 185–198.

