Information Retrivial and Clustering
W. Wu and H. Xiong (Eds.) pp. – - –

Document Clustering, Visualization, and Retrieval via Link Mining

Steven Noel
Center for Secure Information Systems
George Mason University, Fairfax, VA 22030, USA
E-mail: snoel@gmu.edu

Vijay Raghavan
Center for Advanced Computer Studies
The University of Louisiana at Lafayette, Lafayette, LA 70504, USA
E-mail: vraghavan@cacs.louisiana.edu

C.-H. Henry Chu
Center for Advanced Computer Studies
The University of Louisiana at Lafayette, Lafayette, LA 70504, USA
E-mail: cice@cacs.louisiana.edu

Contents

1 Introduction

Clustering for document retrieval has traditionally been done through word-based similarity. But this approach suffers from the ambiguity problems inherent in natural languages. Language-based processing can be augmented by analysis of links among document sets, i.e. hypertext Web links or literature citations. Indeed, early workers in information science recognized the shortcomings with word-based document processing. This led to the introduction of document processing based on literature citations [6]. An important development was the notion of co-citation [13], in which a document pair is associated by being jointly cited (or co-cited) by other documents. In general, clustering based on co-citation as a similarity measure is known to correspond well to document semantics. Spurred by the popularity of the Web, more recent approaches have been developed for analyzing hyperlinks, though primarily for search engine page ranking [12, 8].

Another important recent development in information science is association mining [1]. This measures how strongly certain sets of objects are associated through joint references. Only recently has it been recognized that association mining strength is a generalization of the classical co-citation similarity [11]. In particular, association mining generalizes relationships between pairs of documents to relationships among document sets of arbitrary cardinality. These higher-order associations capture relationships that are generally missed through pairwise co-citation similarities.

Association mining shares a particular aspect of the fundamental clustering hypothesis. Specifically, strongly associated sets are assumed to be composed of highly similar elements. But contrary to the clustering hypothesis, association mining lacks the assumption that objects in strongly associated sets are highly dissimilar to objects outside the set. Unlike the disjoint sets found in clustering, the sets in association mining are overlapping. This can lead to overwhelming numbers of combinations to consider, even for moderate-sized document collections. The challenge is to include higher-order associations while keeping the complexity manageable.

This chapter describes how link association mining can be applied to document clustering. It addresses the fundamental differences between strongly

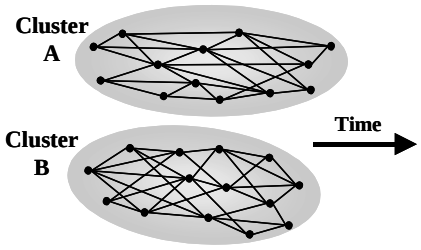

Figure 1: Link-based document clusters.

associated sets and document clusters, as well as how to meet the challenge of retaining higher-order information while maintaining low complexity. The approach described here includes higher-order link association information as similarity features for pairs of documents. From the pairwise document similarities, standard clustering approaches are then applied. In particular, it applies higher-order association similarities to cluster visualization for information retrieval tasks.

2 Link-Based Document Clustering

Traditionally, link-based document analysis has applied co-citation as a similarity measure for clustering. The typical goal was to discover subsets of large document collections that correspond to individual fields of study. Conceptually, documents form clusters if they share links among them in a certain sense, e.g. as shown in Figure 1.

Collections of linked documents (e.g. through citations or hyperlinks) can be modeled as directed graphs, as in Figure 2. A graph edge from one document to another indicates a link from the first to the second. In a matrix formulation, a binary adjacency matrix is formed corresponding to the document link graph. Assume that adjacency matrix rows are for citing (linking-from) documents and columns are for cited (linking-to) documents. Thus for adjacency matrix A, element $a_{i,j} = 1$ indicates that document i cites (or links to) document j, and $a_{i,j} = 0$ is the lack of citation (link).

Co-citation between a pair of documents is the joint citing (or hypertext linking) of the pair by another document, as shown in Figure 3. A traditional measure of similarity between a pair of documents is the number of documents that co-cite the pair, known as citation count. Taken over all pairs of documents, the co-citation count similarity serves as a compact

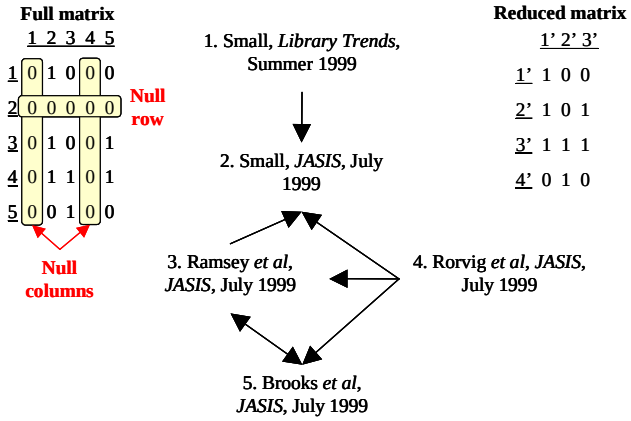

Figure 2: Adjacency matrix for document link graph.

representation of citation graph structure.

Co-citation between a pair of documents is the joint citing (or hypertext linking) of the pair by another document, as shown in Figure 3. A measure of similarity between a pair of documents is the number of documents that co-cite the pair, known as citation count. Taken over all pairs of documents, the co-citation count similarity serves as a compact representation of citation graph structure.

In terms of the document citation adjacency matrix A, co-citation count is a scalar quantity computed for pairs of matrix columns (cited documents). For columns j and k, co-citation count $c_{j,k}$ is then

$$c_{j,k} = \sum_i a_{i,j} a_{i,k} = a_j \cdot a_k = A^T A. \tag{1}$$

Here a_j and a_k are column vectors of A, i indexes rows, A^T is the transpose of A, and $x \cdot y$ is the vector dot (inner) product. Note that the product $a_{i,j} a_{i,k}$ represents single co-citation occurrences, which the summation counts. The co-citation count $c_{j,j}$ of a document with itself is simply a citation count, i.e. the number of times the document has been cited.

It is convenient to normalize the co-citation count $c_{j,k}$ through the linear transformation

$$\hat{c}_{j,k} = \frac{c_{j,k} - \min(c_{j,k})}{\max(c_{j,k}) - \min(c_{j,k})}, \tag{2}$$

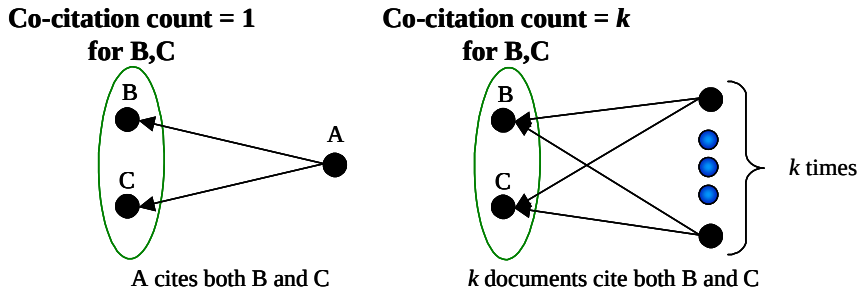

Figure 3: Co-citation document similarity.

yielding the normalized count $\hat{c}_{j,k}\epsilon[0,1]$. Here $\min(\cdot)$ and $\max(\cdot)$ are the minimum and maximum functions, respectively. Standard clustering and minimum spanning tree algorithms assume dissimilarities rather than similarities. We convert similarities to dissimilarities (distances) through the linear transformation

$$d_{j,k} = 1 - \hat{c}_{j,k}. \tag{3}$$

This results in distance $d_{j,k}$ between documents j and k, normalized to $d_{j,k}\epsilon[0,1]$.

Classical co-citation analysis relies on simple single-linkage clustering [16], because of its lower computational complexity given the typically large document collections. But a known problem with single-linkage clustering is a possible "chaining" effect, in which unrelated documents get clustered together through a chain of intermediate documents [14]. Figure 4 shows an example of single-linkage chaining, in which two clusters merge through a single pair of co-cited documents.

Alternative clustering criteria exist that are stronger than the single linkage criterion, e.g. average linkage and complete linkage (see Figure 5). These criteria are applied in an agglomerative clustering heuristic in which clusters that have the closest distance between them are iteratively merged. For single-linkage, the measure of distance between two clusters is the closest possible distance between objects in separate clusters. For average-linkage, cluster distance is the average of distances between objects in separate clusters. For complete-linkage, cluster distance is the furthest distance between objects in separate clusters. Thus single-linkage, average-

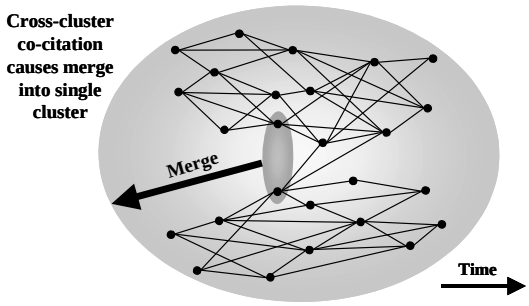

Figure 4: Chaining in co-citation single-linkage clustering.

linkage, and complete-linkage correspond to weak, intermediate, and strong clustering criteria, respectively.

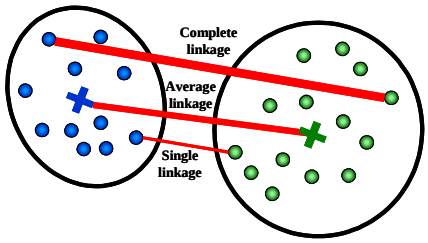

Figure 5: Inter-cluster distances for single-linkage, average-linkage, and complete-linkage.

3 Incorporating Higher-Order Link Information

Citation analysis has traditionally applied single-linkage clustering, because of its lower computational complexity. But being a weak clustering criterion, single-linkage has problems unless the data are inherently well clustered. Given the improved performance of modern computers, it becomes feasible to apply stronger clustering criteria in citation analysis. In fact, we apply a particularly strong criterion taken from the area of association mining.

This involves essentially higher-order co-citations, i.e. co-citations among document sets of arbitrary cardinality.

Beyond providing a stronger clustering criterion, another benefit of higher-order co- citations is with regard to user-oriented clustering. Here the user provides iterative feedback to help guide the clustering process, based on knowledge of the application domain. With pairwise distances, users can orient clustering by weighting distances for various document pairs, applying heavier weights to pairs whose similarities are more important. With higher-order similarities, this orientation can be generalized to weighting document sets of arbitrary cardinality.

3.1 Document Distances from Link Association Mining

Figure 6 illustrates the weak single-linkage criterion, and how it relates to citations. For the three cited documents in the example, there are three possible co-citation similarities (pairs of documents). As the example shows, only two of these similarities need to exceed the clustering threshold for the three documents to be considered a cluster, as long as they share a common document between the two pairs.

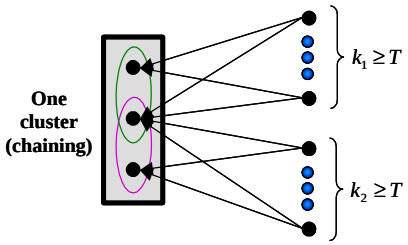

Figure 6: Single-linkage chaining with co-citation similarities.

In contrast, for the stronger clustering criterion of complete linkage all similarities for the three pairs need to exceed the threshold before the documents constitute a single cluster. This is shown in Figure 7. But notice that for this example, there is not even one document that cites all three of the clustered documents simultaneously. The complete- linkage criterion is a necessary but not sufficient condition for the simultaneous citing of all documents in a cluster.

But consider a generalization of co-citation similarity in which sets of arbitrary cardinality are considered for co-citation, as shown in Figure 8.

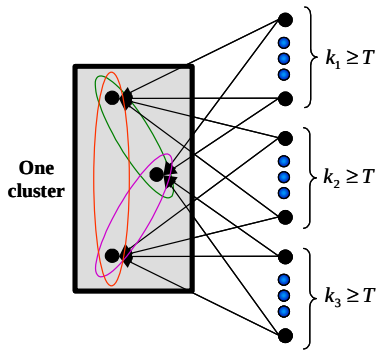

Figure 7: Stronger complete-linkage criterion with co-citation similarities.

That is, we define similarity as the number of times all the members of the set are simultaneously cited. Because the similarity involves more than two documents, it is higher order than pairwise similarity.

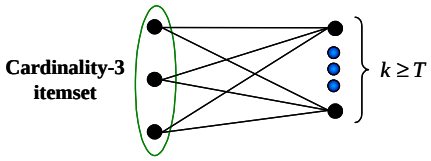

Figure 8: Itemset is an even stronger association than complete-linkage cluster.

We can specify a threshold value for these higher-order similarities (itemset supports) to identify sets whose similarities are sufficiently large. For our example, the only way the three cited documents could be considered a sufficiently similar set is if all three of them are cited more than the threshold number of times. In association mining, such a jointly referenced set is known as an *itemset*. The number of times that the set is jointly referenced (co-cited) is known as the itemset *support*. Itemsets whose members are sufficiently similar (have sufficient support) are known as *frequent* itemsets.

The extension from pairs of documents to sets of arbitrary cardinality means there is itemset overlap, that is, itemsets are non-disjoint. Such over-

lap is not possible with pairs of documents. Itemset supports of arbitrary cardinality are thus represented as *lattices* rather than $n \times n$ matrices for n documents.

In particular, itemsets are represented by the lattice of all subsets of a document collection. The subsets form a partial ordering, under the ordering relation of set inclusion. This is illustrated in Figure 9, via the Hasse diagram for visualizing partial orderings. The diagram shows the itemset lattice (excluding singletons and the empty set) for a set of four documents.

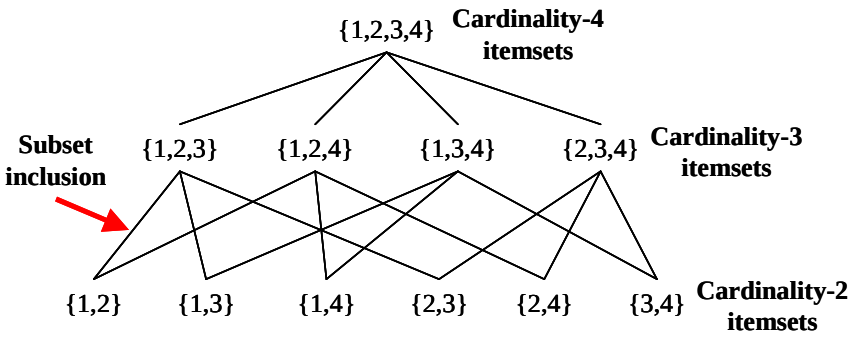

Figure 9: Itemset lattice for a set of 4 documents.

Itemset cardinality corresponds to a single level of the Hasse diagram. For itemset cardinality $|I|$, the number of possible itemsets is

$$\left(\begin{array}{c} n \\ |I| \end{array} \right) = \frac{n!}{|I|!(n-|I|)!}, \tag{4}$$

for n documents. The total number of possible itemsets over all cardinalities is 2^n.

In our matrix formalism, itemset supports are computed for sets of columns (cited documents) of the adjacency matrix, just as they are computed for pairs of columns in computing co-citation counts. For itemset I of cardinality $|I|$, whose member documents correspond to columns $j_1, j_2, \cdots, j_{|I|}$, its scalar support $\zeta(I)$ is

$$\zeta(I) = \sum_i a_{i,j_1} a_{i,j_2} \cdots a_{i,j_{|I|}} = \sum_i \prod_{\alpha=1}^{|I|} a_{i,j_\alpha}, \tag{5}$$

where i indexes rows (citing documents). Just as for pairwise co-citations, the term $a_{i,j_1}a_{i,j_2}\cdots a_{i,j_{|I|}}$ represents single co-citation occurrences, which are now generalized to higher orders. The summation then counts the individual co-citation occurrences.

A central problem in data mining is the discovery of frequent itemsets, i.e. itemsets with support greater than some threshold amount. In the context of linked documents, such frequent itemsets represent groups of highly similar documents based on higher- order co-citations. But managing and interacting with itemsets for document retrieval is problematic. Because of the combinatorially exploding numbers of itemsets and their overlap, user interaction becomes unwieldy.

Also, standard tools of analysis and visualization such as clustering and the minimum spanning tree assume an input matrix of pairwise distances. Mathematically, distances for all document pairs correspond to a fully connected distance graph. But the generalization to higher-order distances means that the distance graph edges are generalized to hyperedges, that is, edges that are incident upon more than two vertices. It is difficult to generalize clustering or minimum spanning tree algorithms to such distance hypergraphs.

We solve this dilemma by applying standard clustering or minimum spanning tree algorithms, but with pairwise distances that include higher-order co-citation similarities. The new distances we propose are thus a hybrid between standard pairwise distances and higher-order distances. For information retrieval visualization, users need only deal with disjoint sets of items, rather than combinatorial explosions of non-disjoint itemsets. The approach is designed such that member documents of frequent itemsets are more likely to appear together in clusters.

The following is the chain of reasoning leading to these link-mining distances. Consider the lattice of itemsets partially ordered by the set inclusion relation. Associated with each itemset in the lattice is its support. Consider any cardinality-2 itemset (document pair) $\{j,k\}$. Itemsets I that are proper supersets of $\{j,k\}$ are greater than $\{j,k\}$ in terms of the partial ordering, that is, $\{j,k\} \subset I \Leftrightarrow \{j,k\} < I$.

All the information at our disposal about the pairwise distance $d(j,k) = d(k,j)$ is then contained in the supports for these itemsets: $\{j\}$, $\{k\}$, $\{j,k\}$, and all itemsets $I > \{j,k\}$. The traditional way to compute the distance $d(j,k)$ is based simply on the support of $\{j,k\}$ as a measure of similarity. After normalization, the support is converted from a similarity to a dissimilarity (distance) via multiplicative or additive inversion.

A straightforward model for higher-order document similarities is to

simply sum supports over all itemsets that contain the document pair in question. The itemset supports can thus be interpreted as features. More formally, the itemset support feature summation is

$$s_{j,k} = \sum_{\{I:j,k\epsilon I\}} \zeta(I). \tag{6}$$

This yields the similarity $s_{j,k}$ between documents j and k, where $\zeta(I)$ is the support of itemset I.

It is not feasible to compute all possible itemset supports. In practice, a reasonable approximation is to compute only those itemsets above a certain minimum support $\zeta_{\min}$. The document similarity in (6) then becomes

$$s_{j,k} = \sum_{\{I:j,k\epsilon I;\zeta(I)\geq\zeta_{\min}\}} \zeta(I). \tag{7}$$

By doing this, we can take advantage of fast algorithms (e.g. the *Apriori* algorithm [2]) for computing so-called frequent itemsets.

Algorithms for computing frequent itemsets are generally based on two principles. The first is that every subset of a frequent itemset is also frequent, so that higher- cardinality itemsets need only be considered if all their subsets are frequent. The second is that given a partitioning of the set of citing documents, an itemset can be frequent only if it is frequent in at least one partition, allowing the application of divide-and-conquer algorithms.

The worst-case complexity of the frequent-itemset problem is exponential. But empirically, fast algorithms scale linearly with respect to both the number of citing documents and the number of documents each of these cite. In fact, empirically these algorithms actually speed up slightly as the total size of the cited document collection increases.

If (7) is applied directly as a clustering similarity, experience has shown that the resulting clusters match poorly with frequent itemsets. In other words, this similarity formula is poor at preserving the higher-order co-citations (citation frequent itemsets). But an extension of this formula has been proposed [11] that yields clusters that are generally consistent with frequent itemsets.

In particular, a nonlinear transformation $T[\zeta(I)]$ is applied to the itemset supports $\zeta(I)$ before summation. The transformation T must be super-linear (asymptotically increasing more quickly than linearly), so as to favor large itemset supports. Example transformations include raising to a power greater than unity, or an increasing exponential. The document similarity

in (7) then becomes

$$s_{j,k} = \sum_{\{I:j,k\epsilon I;\zeta(I)\geq\zeta_{\min}\}} T[\zeta(I)]. \tag{8}$$

The similarity $s_{j,k}$ is then normalized via

$$\hat{s}_{j,k} = \frac{s_{j,k} - \min(s_{j,k})}{\max(s_{j,k}) - \min(s_{j,k})}, \tag{9}$$

yielding the normalized similarity $\hat{s}_{j,k}\epsilon[0,1]$. Finally, this is converted from similarity to distance via additive inversion:

$$d_{j,k} = 1 - \hat{s}_{j,k}. \tag{10}$$

The resulting distance formula has the properties we desire. In particular, it includes higher-order co-citation information (association mining itemset supports), while retaining a simple pairwise structure. This provides a much stronger link-based clustering criterion, while enabling the direction application of standard similarity-based clustering algorithms.

3.2 Example Application of Link-Mining Distances

Let us examine an example application of the document distances in (10). We take this example from a set of documents retrieved from the Science Citation Index [18], which contains citation links among all documents in its collection. The query for this example is keyword "microtubules" for year 1999.

We select the first 100 documents returned from the query, which together cite 3070 unique documents. We retain only those 45 documents that have been cited 6 or more times, to retain only the more frequently cited documents. Figure 10 shows the resulting 100×45 citation matrix. In the figure, black indicates presence of citation and white indicates lack of citation.

Figure 11 shows a distribution of itemset supports for the "microtubules" document set. In particular, the left-hand side shows the support for each 3-itemset (itemset of cardinality 3), and the right-hand side shows the log-histogram of those supports. Cardinality-3 itemsets were chosen as representative of the general form that itemset distributions take. In fact, this support distribution is representative not only of itemsets of a particular cardinality, but for general document sets as a whole.

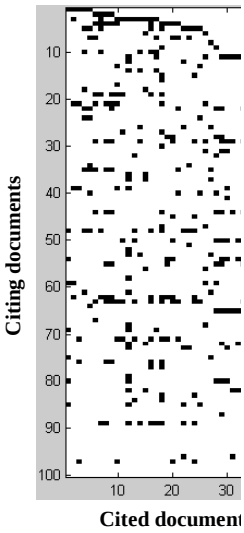

Figure 10: Citation matrix for SCI "microtubules" document set.

Figure 11 also includes the positive side of the Laplacian distribution, which appears linear on the log scale. The general Laplacian distribution $p(x)$ is

$$p(x) = \frac{\lambda}{2} e^{-\lambda|x|}. \tag{11}$$

The distribution is parameterized by $\lambda = \sqrt{(}2)/\sigma$, for standard deviation σ. Since itemset support is always positive, only the positive side of the distribution is needed, i.e.

$$p^{+}(x) = \lambda e^{-\lambda x}, x \geq 0. \tag{12}$$

As in Figure 11, Figure 12 shows cardinality-3 itemset supports for the "microtubules" document set. But this time, the itemset supports are non-linearly transformed, and only frequent itemsets (those with support above a particular threshold) are included. This shows the actual itemset support values that appear in the document similarity computation of (8).

It is important to point out the effect that the Laplacian itemset support distribution has on the computational complexity of our approach. Consider that fast algorithms for frequent itemsets compute supports starting from

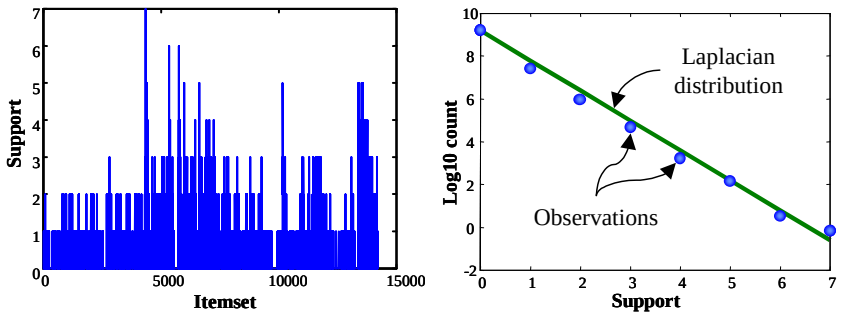

Figure 11: Itemset-support distribution for "microtubules" document set.

lower cardinalities. In going to the next higher cardinality, they consider only supersets of those itemsets found to be frequent at the lower cardinality. Because of the very heavy concentration of smaller supports in the Laplacian distribution, a large portion of itemsets are below the minimum support threshold, and get removed from any further consideration at higher cardinalities. This in turn contributes to low average computational complexity.

The Laplacian distribution of itemset supports is also consistent with the fact that our approach yields clusters that tend to be consistent with more frequent itemsets. Frequent itemsets are already sparse for this distribution, and the nonlinear transformation in (8) increases the sparseness. In fact, we can give a theoretical guarantee for the consistency between clusters and frequent itemsets. This guarantee holds independent of any particular clustering criterion.

Details of the proof are in [10]. The key idea of the proof is that for distances computed with (8) through (10), the most frequent itemset can always be made a cluster, given a large enough degree of nonlinearity in the transformation of itemset supports. This relies on the fact that for a super-linear transformation $T = T_p = T(\zeta; p)$ of itemset supports ζ in (8), as the degree of nonlinearity p increases, $T_p(\zeta)$ with a larger ζ is asymptotically bounded from below by $T_p(\zeta)$ with a smaller ζ. Since the term with largest ζ asymptotically dominates the distance summation, the result is that documents in the most frequent itemset are asymptotically closer to one another than to any other documents, thus forming a cluster.

We then generalize the proof to cover the clustering of arbitrary itemsets

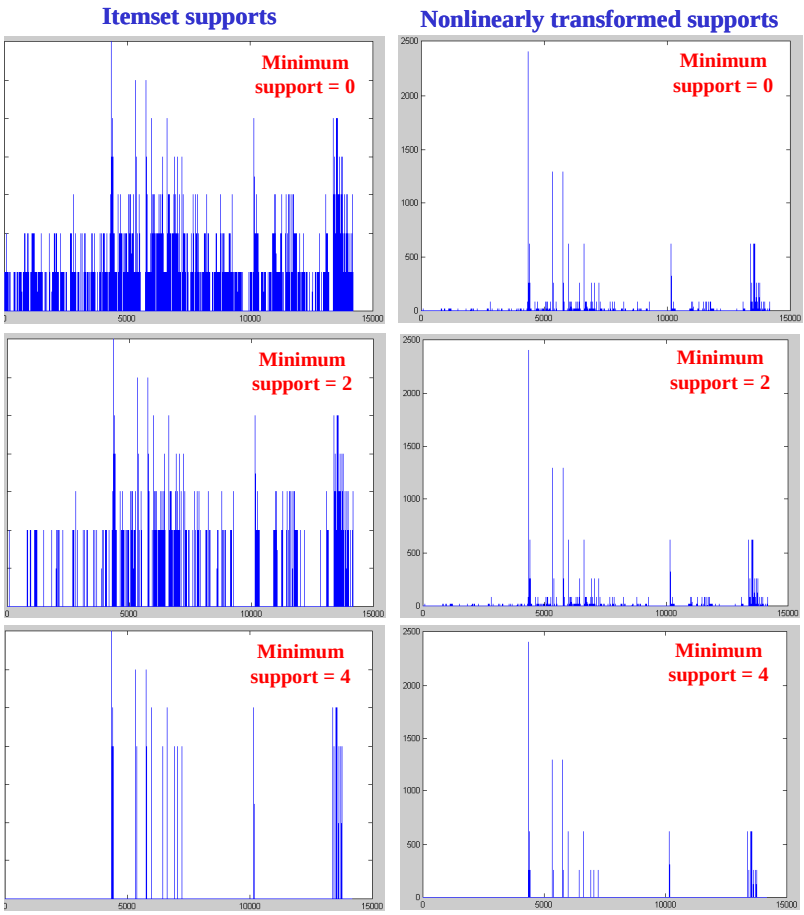

Figure 12: Transformed frequent itemsets for "microtubules" document set.

in terms of their relative supports and document overlap. The result is that more frequent itemsets asymptotically form clusters at the expense of less frequent itemsets that overlap them. If there is no overlapping itemset with more support, then a given itemset will form a cluster for a sufficiently large value of the nonlinearity parameter p. For overlapping itemsets with equal support, the itemset members not in common asymptotically form clusters, and each of the members in common are asymptotically in one of those clusters, though there is no guarantee for exactly which of them.

Our theoretical result provides no such guarantee for itemsets of equal support. More importantly, it provides no upper bound on the necessary value of p to ensure itemset clustering for a given data set. But empirically, we have found [11] that the transformation

$$T(\zeta) = \zeta^p \tag{13}$$

with $p = 4$ usually results in the most frequent itemsets appearing together in clusters.

4 Link Mining for Hierarchical Document Clustering

For document retrieval, results from simple keyword queries can be clustered into more refined topics. Such clustering can help users identify subsets of the retrieval results that are more likely to contain relevant documents. Indeed, studies have shown that precision and recall tends to be higher within the best cluster than within the retrieval results as a whole [19]. Users can thus concentrate on clusters with higher proportions of relevant documents, avoiding clusters with mostly non-relevant documents. Users can search for better clusters by looking at small samples from each cluster.

Additionally, document clustering can be applied hierarchically, yielding clusters within clusters. This facilitates a large-to-small-scale analysis of retrieval results reminiscent of binary search. This is based on the idea that larger scale clusters correspond to more general topics, and smaller scale clusters correspond to more specific topics within the general topics. Cluster hierarchies thus serve as topic hierarchies. For information retrieval, content analysis of document samples within large-scale clusters allows the user to choose among general topics. Once a particular general topic is chosen, the user can sample documents within smaller-scale clusters within the large- scale cluster. The user can then proceed with this analysis recursively, progressively refining the topic.

4.1 Hierarchical Clustering, Dendrograms, and Document Retrieval

The dendrogram is a tree visualization of a hierarchical clustering. Leaves of the dendrogram tree are individual documents, at the lowest level of the hierarchy. Non-leaf nodes represent the merging of two or more clusters, at increasing levels of the hierarchy. A node is drawn as a horizontal line that spans over its children, with the line drawn at the vertical position corresponding to the merge threshold distance. The dendrogram visualization is illustrated in Figure 13.

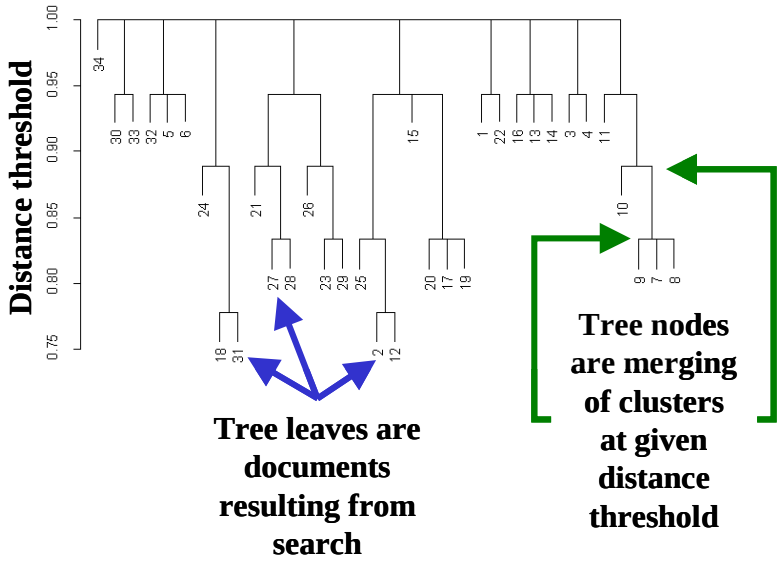

Figure 13: Dendrogram for visualizing hierarchical document clusters.

For a given hierarchical clustering, the clusters resulting from a given threshold distance can be readily determined from the dendrogram (see Figure 14). If a horizontal line is envisioned at the given threshold value, tree nodes directly below the threshold define clusters. That is, the nodes' children are all members of the same cluster. Nodes that lie above the threshold each represent clusters of single documents.

The dendrogram serves as the basis for a graphical user interface for information retrieval. The hierarchical display facilitates interactive large-

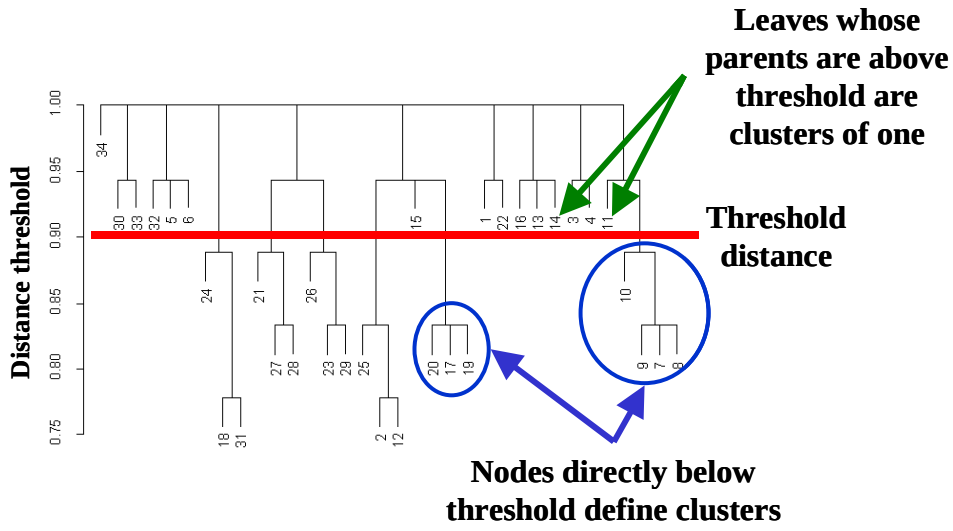

Figure 14: Interpreting clusters from dendrogram for given clustering threshold distance.

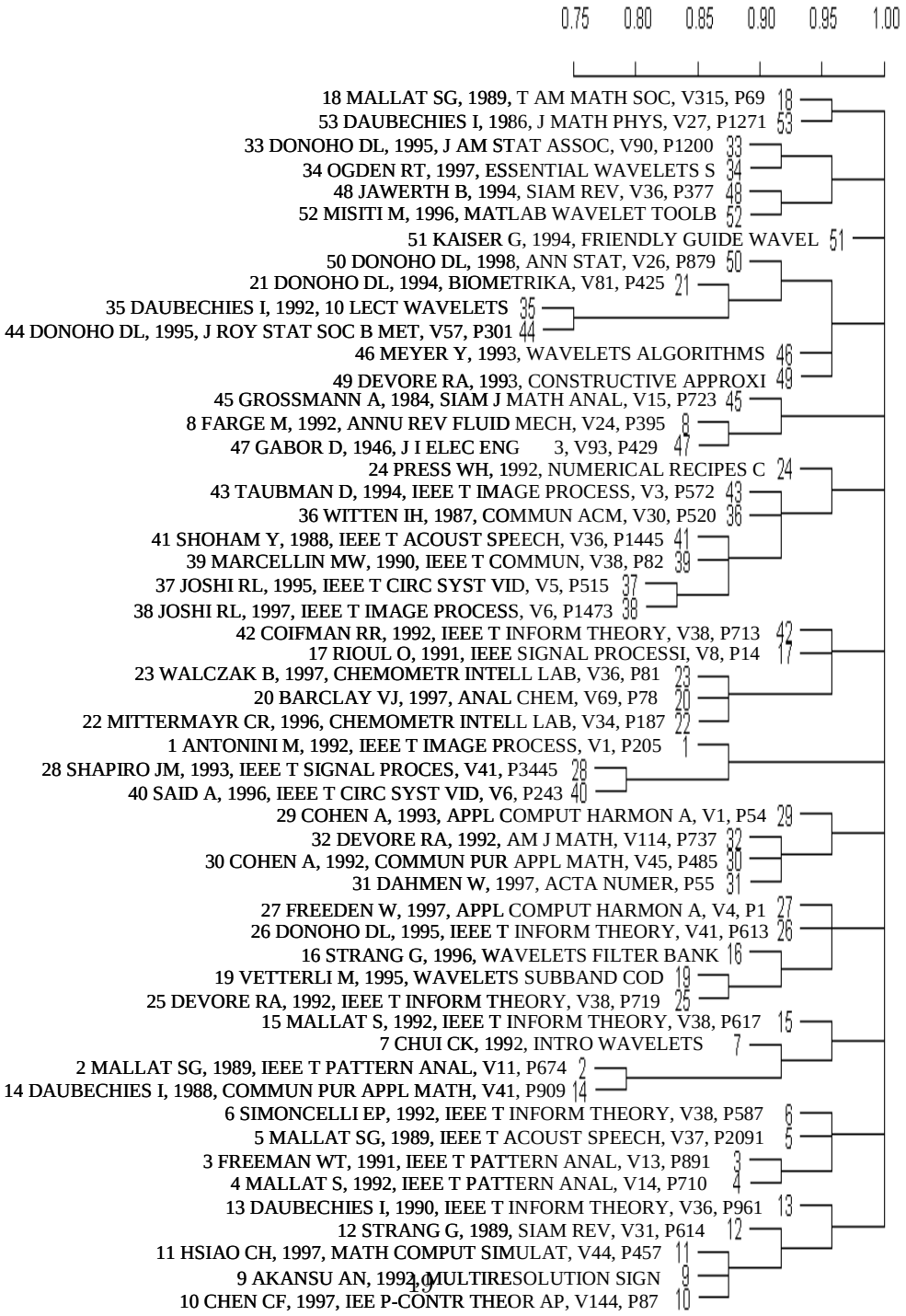

Figure 15: Dendrogram rotated and augmented with text for document retrieval.

to-small-scale cluster analysis for topic refinement. By selecting a leaf node, the user can retrieve metadata for an individual document, or can retrieve the document itself.

For a document retrieval user interface, the dendrogram can be rotated by 90 degrees and augmenting with text, as shown in Figure 15. In the usual dendrogram orientation, there is an arbitrary amount of blank space available below each leaf, in the vertical direction. The 90-degree rotation provides arbitrary space available in the horizontal direction, consistent with lines of text. Within information retrieval, the general approach of clustering for topic refinement is well established. But for document searching, some questions have been raised about the value added by clustering visualizations in comparison to the traditional query approach [19].

Clustering visualizations have obvious value in providing thematic overviews of document collections, particularly useful for browsing. But for the search process, the lack of textual representations such as titles and topical words can make systems difficult to use. In general, it is difficult to understand document contents without actually reading some text. The situation is compounded by the fact that it is difficult to objectively measure visualization effectiveness [7].

4.2 Illustrative Example

We now demonstrate our approach to link-based document clustering for an example document set. In particular, we compare link association mining results to clusters resulting from our link-based similarity formula (8). The example document set is from the Science Citation Index (SCI) [18].

We do the itemset/clustering comparison by a novel augmentation of the dendrogram with members of frequent itemsets. This allows an easy visual assessment of itemset/cluster consistency. For this example, we do an SCI query with keyword "wavelet*" for the year 1999. The first 100 documents returned by the query cite 1755 documents. We filter these cited documents by citation count, retaining only those cited three or more times, resulting in a set of 34 highly cited documents. We then compute complete-linkage, average-linkage, and single-linkage clusters for the set of 34 highly cited documents. Here we first apply the traditional pairwise method of computing co-citation based distances. The resulting augmented dendrogram is shown in Figure 16. The dendrogram is augmented by the addition of graphical symbols for members of frequent 4-itemsets, added at the corresponding tree leaves. In this example, the most frequently occurring 4-itemset is {2, 17, 19, 20} . For complete linkage, documents 17, 19, and 20 of this itemset

are a possible cluster. These documents apply wavelets to problems in the field of chemistry, and are well separated from the rest of the collection, both thematically and in terms of co-citations. But including document 2 (a foundational wavelet paper by wavelet pioneer Mallat) in this cluster would require the inclusion of documents 12, 15, and 25, which are not in the itemset. These three additional documents are another foundational wavelet paper by Mallat, and two foundational papers by fellow pioneer Daubechies.

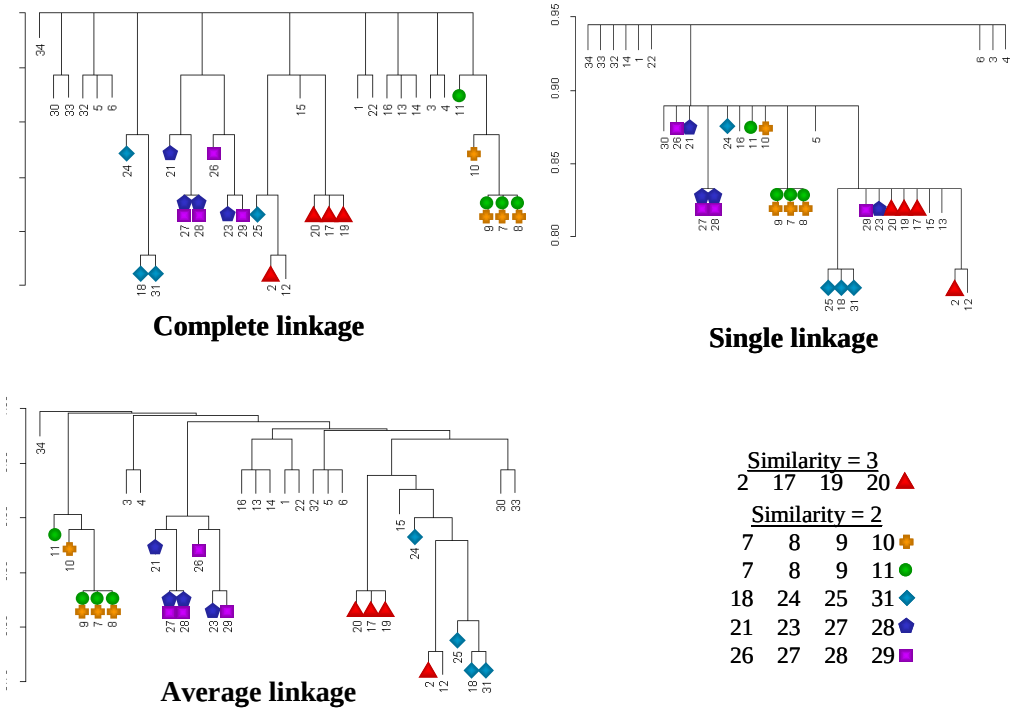

Figure 16: Inconsistency between clusters and frequent itemsets for pairwise document distances.

For single linkage, there is even less cluster/itemset consistency. The itemset {2, 17, 19, 20} is possible within a cluster only by including 8 other documents. We interpret this as being largely caused by single linkage chaining. In general, the application of clustering to mere pairwise co-citation similarities is insufficient for ensuring that itemsets of larger cardinality ap-

pear as clusters, even with complete-linkage. The overlap for the 4-itemsets {7, 8, 9, 10} and {7, 8, 9, 11} , corresponds to the 5-itemset {7, 8, 9, 10, 11}. Thematically, these 5 papers are largely foundational. The combined two 4-itemsets are a complete-linkage cluster. But for single-linkage, 24 other documents would need to be included in order for the two itemsets to be a cluster. Again, pairwise clustering is a necessary but insufficient condition for frequent itemsets. We have a similar situation for the 4-itemsets {21, 23, 27, 28} and {26, 27, 28, 29} , though with a lesser degree of itemset overlap. Thematically, these papers are applications of wavelets in image coding.

For the 4-itemset {18, 24, 25, 31} , three of the papers are by Donoho, who works in wavelet-based statistical signal estimation for denoising. These three papers are a complete-linkage cluster, as well as a single-linkage cluster. The remaining document in the 4-itemset is a foundational book by Daubechies. Including it in a complete-linkage cluster would require the inclusion of every document in the set, while including it in a single-linkage cluster would require the inclusion of 21 other documents.

As a comparison with traditional pairwise co-citation clustering, Figure 17 shows clusters for our distances that include link-mining itemset supports. In particular, it shows complete-linkage clusters with document distances computed via (8) through (10). There are three separate cases, each case being taken over multiple values of itemset cardinality χ. The three cases are $\chi = 2, 3$; $\chi = 2, 3, 4$; $\chi = 3, 4$. Here the itemset supports $\zeta(I)$ are nonlinearly transformed by $T[\zeta(I)] = [\zeta(I)]^4$.

Consistency between clusters and frequent itemsets is nearly perfect with our link-mining distances. The most frequent itemset {2,17,19,20} forms a cluster for two of the cases ($\chi = 2, 3, 4$ and $\chi = 3, 4$). The source of the inconsistency for the case $\chi = 2, 3$ is apparently the lowest-order (pairwise) supports. Lower order supports are generally larger than higher-order supports, and thus tend to dominate the summation in (8). All other frequent itemsets are consistent with these clusters, at least to the extent possible given their overlap. That is, {7, 8, 9, 10} overlaps with {7, 8, 9, 11} and {21, 23, 27, 28} overlaps with {26, 27, 28, 29}, which prevents them from forming individual clusters.

4.3 Itemset-Matching Clustering Metric

In comparing clustering to association mining itemsets, the important issue is whether frequent itemsets form clusters comprised only of the itemset members. This is equivalent to determining the minimal-cardinality cluster that contains all the members of a given itemset and then comparing that

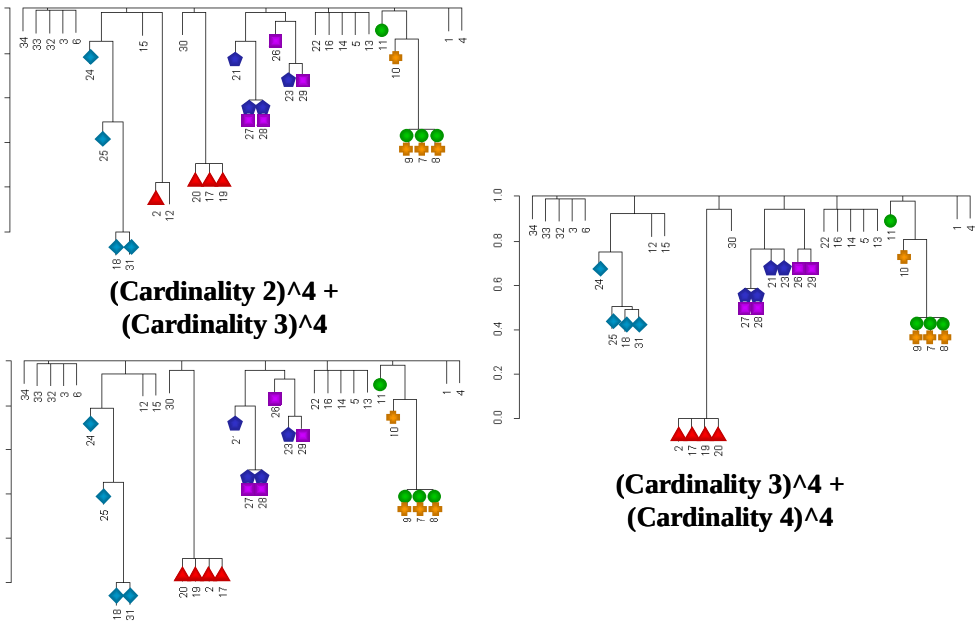

Figure 17: Clusters from our link-mining distances are much more consistent with frequent itemsets.

cluster cardinality to the itemset cardinality. The portion of a minimal cluster occupied by an itemset could serve as an itemset-matching metric for a clustering. Moreover, it could be averaged over a number of itemsets to yield an overall itemset-matching metric for a clustering.

We describe this itemset-matching metric more formally. Let

$$\pi = \{\pi_1, \pi_2, \cdots, \pi_{k_1}\}$$

be a partition of items (documents) that is consistent with a hierarchical clustering merge tree. Furthermore, let $I = \{I_1, I_2, \cdots, I_{k_2}\}$ be a set of itemsets. Then for each itemset $I_i \epsilon I$, there is some block of the partition $\pi_j \epsilon \pi$ such that $|\pi_j|$ is minimized, subject to the constraint that $I_i \subseteq \pi_j$. We call this π_j the minimal cluster containing the itemset. The fact that such a minimal cluster exists can be proven by straightforward induction. The constraint $I_i \subseteq \pi_j$ is satisfied trivially for a partitioning in which a single block contains all items in the original set, corresponding to the highest level of the merge tree.

Moving down to the next highest level of the merge tree, either some block of the partition $\pi_j \epsilon \pi$ satisfies $I_i \subseteq \pi_j$, or else not. If not, then the block in the highest-level partition is the minimal cluster containing the itemset. Otherwise this process can be repeated, until a level is reached in which the constraint $I_i \subseteq \pi_j$ fails. At this point, the minimal cluster containing the itemset is found from the previous level, as the one in which $I_i \subseteq \pi_j$. A similar argument can start from the leaves of the merge tree and proceed upward.

Once a minimal (cardinality) cluster π_j is found for an itemset, a metric can be defined for measuring the extent to which the itemset is consistent with the cluster. This metric $M(\pi, I_i)$ is simply the portion of the cluster occupied by the itemset, or in terms of set cardinalities,

$$M(\pi, I_i) = \frac{|I_i|}{|\pi_j|}.$$

Again, this requires that $|\pi_j|$ be minimized, for $\pi_j \epsilon \pi$, subject to the constraint $I_i \subseteq \pi_j$, and π is consistent with the merge tree. The metric $M(\pi, I_i)$ is defined for a set of itemsets I by averaging $M(\pi, I_i)$ over $I_i \epsilon I$, that is,

$$M(\pi, I) = \frac{1}{|I|} \sum_{I_i \epsilon I} M(\pi, I_i) = \frac{1}{|I|} \sum_{I_i \epsilon I} (\frac{|I_i|}{|\pi_j|}). \tag{14}$$

The itemset-matching metric $M(\pi, I)$ takes its maximum value of unity when $I_i = \pi_j$, indicating the best possible match between itemsets and

clusters. The proof is that since $|I_i| = |\pi_j|$,

$$M(\pi, I) = \frac{1}{|I|} \sum_{I_i \epsilon I} 1 = \frac{|I|}{|I|} = 1.$$

The minimum value of $M(\pi, I)$ is $M(\pi, I) = |I_i|/n$, indicating the poorest possible match. For the proof, consider that $M(\pi, I) = |I_i|/|\pi_j|$ for a given $|I_i|$ takes its minimum value when $|\pi_j|$ takes its maximum value of $|\pi_j| = n$. Then the minimum $M(\pi, I)$ is the sum of minimum $M(\pi, I_i)$, that is,

$$M(\pi, I) = \frac{1}{|I|} \sum_{I_i \epsilon I} (\frac{|I_i|}{|\pi_j|}) = \frac{1}{|I|} \sum_{I_i \epsilon I} (\frac{|I_i|}{n}) = \frac{1}{|I|} \frac{|I_i|}{n} \sum_{I_i \epsilon I} 1 = \frac{|I|}{|I|} \frac{|I_i|}{n} = \frac{|I_i|}{n}.$$

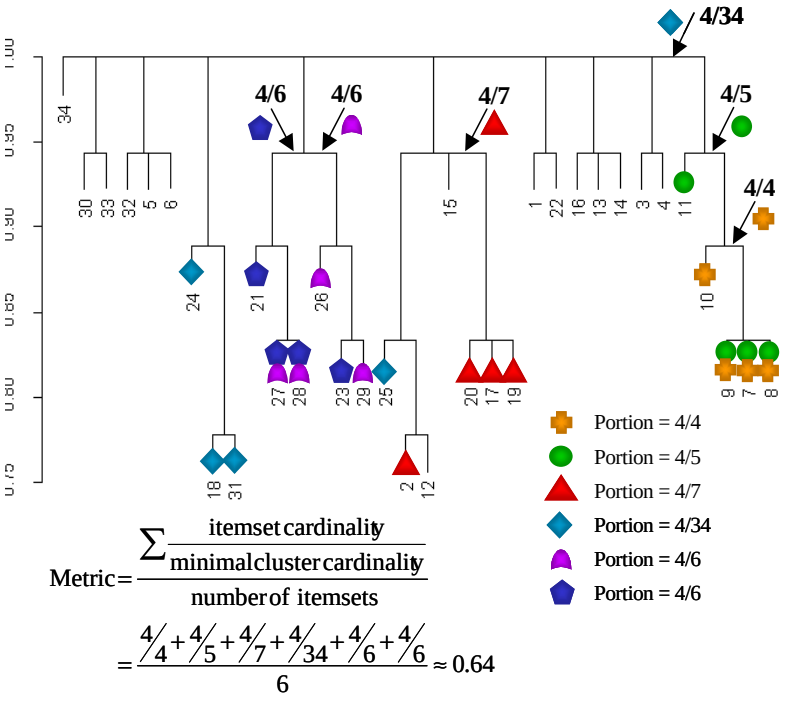

Figure 18: Itemset-matching clustering metric.

Figure 18 illustrates the itemset-matching clustering metric $M(\pi, I)$. For a given itemset, there is some minimal threshold value at which it is a subset

(not necessarily proper) of a cluster. For this threshold value, the cluster may contain documents other than those in the itemset (in which case the itemset is a proper subset of the cluster). The ratio of the itemset cardinality to cluster cardinality is the size of the itemset relative to the cluster size. The metric is then the average of these relative itemset sizes over a set of itemsets.

4.4 Experimental Validation

In this section, we apply our proposed document distances to real-world document citations. In particular, we apply them to data sets from the Science Citation Index (SCI). Science citations are a classical form of hypertext, and are of significant interest in information science. The assumption that citations imply some form of topical influence holds reasonably well. That is, for citations all links between documents have the same general semantics. This is in contrast to Web documents, in which links could be for a variety of other purposes, such as navigation. The general method of clustering documents and computing itemset-matching clustering metrics is shown in Figure 19. We include higher-order co-citations in pairwise document distances, and compute standard co-citation distances as a comparison. We also compute frequent itemsets of given cardinalities and minimum supports. Once a clustering is computed for the given distance function, we subject it to the itemset-matching metric for the given itemsets.

The SCI data sets we employ are described in Table 1. For each data set, the table gives the query keyword and publication year(s), the number of citing documents resulting from the query, and the number of documents they cite after filtering by citation count. For the data sets 1, 5, and 9, results are included for both co-citations and bibliographic coupling, yielding data sets 2, 6, and 10 (respectively), for total of 10 data sets.

Our empirical tests apply the metric proposed in Section 4.3. The metric compares clustering to frequent itemsets, determining whether given itemsets form clusters comprised only of the itemset members. In other words, the metric determines the minimal-cardinality cluster that contains all the members of a given itemset, and compares that cluster cardinality to the itemset cardinality. This is then averaged over a number of itemsets, to yield an overall itemset-matching metric for a clustering.

Table 2 shows how link-mining distances are computed for the experiments with SCI data sets. The table shows the itemset cardinalities $|I|$ that are applied in the similarity formula (2), and the values of itemset support nonlinearity parameter p for itemset nonlinearity $T(\zeta) = \zeta^p$. For each link-

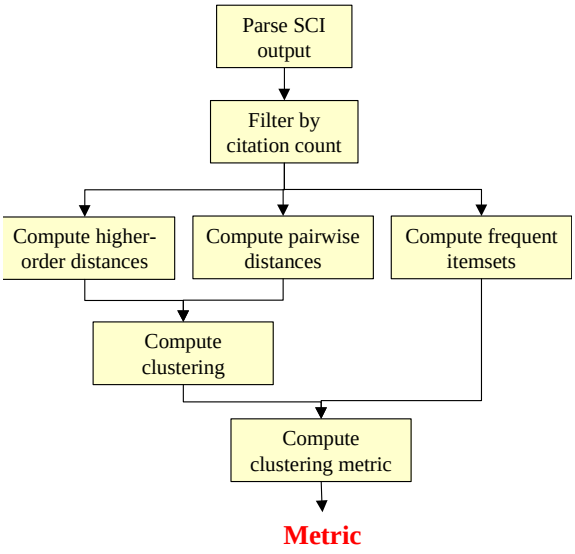

Figure 19: General method for computing itemset-matching clustering metric.

Table 1: Details for Science Citation Index data sets.

Data Set(s)	Query Keyword(s)	Year(s)	Citing Docs	Cited Docs
1, 2	adaptive optics	2000	89	60
3	collagen	1975	494	53
4	genetic algorithm* AND neural network*	2000	136	57
5,6	quantum gravity AND string*	1999-2000	114	50
7	wavelet*	1999	100	34
8	wavelet*	1999	472	54
9,10	wavelet* AND brownian	1973-2000	99	59

mining distance formula, we compare metric values to those for standard pairwise distances. Here we apply the same cardinalities in the metric formula (14) as in the distance formula (8), to enable a direct interpretation of the results.

Table 2: Itemset cardinalities and support nonlinearities for link- mining distances.

Data Set(s)	[Itemset Cardinality, Support Nonlinearity]
1	[3,4], [3,6], [4,4], [4,6]
2,6	[3,4], [4,4], [4,6]
3,5,7,8,9,10	[3,4], [4,4]
4	[3,4], [3,6], [4,4]

The comparisons are done for each combination of complete-linkage, average- linkage, and single-linkage clustering. We also compare for each combination of the most frequent itemset, the 5 most frequent itemsets, and the 10 most frequent itemsets in the metric formula (14). For test cases in which nonlinearity parameter value $p = 4$ yields relatively low metric values, we include additional test cases with $p = 6$. Since there are 25 different combinations of itemset cardinalities and support nonlinearities, each with $3 \times 3 = 9$ different combinations of clustering criteria and sets of most frequent itemsets, there are $9 \times 25 = 225$ total test cases.

Table 3 compares clustering metric values for the test cases in Table 2. The table classifies cases as having metric values for link-mining distances equal to, greater than, or less than metric values for standard pairwise distances. Higher metric values correspond to clusters being more consistent with frequent itemsets. For most test cases (169 of 225 versus 27 of 225), the link-mining distances result in better consistency with frequent itemsets in comparison to standard pairwise distances. For a relatively small number of cases (29 of 225), the metric values are equal.

For the 20 combinations of 10 data sets and 2 itemset cardinalities, we find that itemset support nonlinearities with $p = 4$ are usually sufficient (15 of 20) for a good match to frequent itemsets. Otherwise nonlinearities with $p = 6$ in (13) are sufficient, in all but one of the 20 combinations. Here we consider a clustering metric value greater than about 0.7 to be a good match. This corresponds to a frequent itemset comprising on average about 70% of a cluster that contains all its members.

For data sets 3, 5, 8, 9, and 10, we compute itemset-matching cluster-

Table 3: Clustering metric comparisons for standard pairwise (P.W.) vs. link-mining higher-order (H.O.) distances.

Data set	H.O. = P.W.	H.O. > P.W.	H.O. < P.W.	Cases
1	6	16	14	36
2	7	15	5	27
3	0	18	0	18
4	1	24	2	27
5	3	13	2	18
6	2	22	3	27
7	2	16	0	18
8	5	13	0	18
9	3	14	1	18
10	0	18	0	18
Total	29	169	27	225

ing metrics resulting from the complexity reduction technique in (7), and compare metric values to the full-complexity method in (6). The clustering metric results of the 90 cases from the five extracted data sets are summarized in Table 4 and Table 5. The results show that excluding itemset supports below itemset minimum support *minsup* generally has little effect on clustering results, particular for smaller values of *minsup*. However, there is some degradation in metric values for higher levels of *minsup*. Here "degradation" means that metric values are smaller when some itemset supports are excluded, corresponding to a poorer clustering match to frequent itemsets.

We offer the following interpretation for the *minsup*-dependent degradation in clustering metric. Members of frequent itemsets are typically frequently cited documents overall. Such frequently cited documents are likely to appear in many itemsets, even less frequent itemsets. Thus there are likely to be many itemsets below *minsup* that contain these frequently cited documents. Excluding itemsets below *minsup* then removes the supports that these itemsets contribute to the summations in computing link-mining distances.

Table 4: Clustering metrics for link-mining distances with full computational complexity (minsup 0) and reduced complexity (minsup 2).

Data set	(minsup 2) = (minsup 0)	(minsup 2) > (minsup 0)	(minsup 2) < (minsup 0)	Cases
3	18	0	0	18
5	18	0	0	18
8	18	0	0	18
9	18	0	0	18
10	11	0	7	18
Total	83	0	7	90

Table 5: Clustering metrics for link-mining distances with full computational complexity (minsup 0) and reduced complexity (minsup 4).

Data set	(minsup 4) = (minsup 0)	(minsup 4) > (minsup 0)	(minsup 4) < (minsup 0)	Cases
3	12	2	4	18
5	11	1	6	18
8	10	0	8	18
9	18	0	0	18
10	12	2	4	18
Total	63	5	22	90

5 Conclusions

In this chapter, we described new methods for enhanced understanding of relationships among documents that are returned by information retrieval systems based on a link or citation analysis. A central component in the methodology is a new class of inter-document distances that includes information mined from a hypertext collection. These distances rely on a higher-order counterparts of the familiar co-citation similarity, in which co-citation is generalized from a relationship between a pair of documents to one between arbitrary numbers of documents. These document sets of larger cardinality are equivalent to itemsets in association mining. Our experimental results show that in comparison to standard pairwise distances, our higher-order distances are much more consistent with frequent itemsets.

We also presented the application of the hierarchical clustering dendrogram for information retrieval. The dendrogram enables quick comprehension of complex query-independent relationships among the documents, as opposed to the simple query-ranked lists usually employed for presenting search results. We introduced new augmentations of the dendrogram to support the information retrieval process, by adding document-descriptive text and glyphs for members of frequent itemsets.

This work represents the original application of association mining in finding frequent itemsets for the purpose of visualizing hyperlink structures in information retrieval search results. The generalization of co-citation to higher orders helps prevent the obscuring of important frequent itemsets that often occurs with traditional co-citation based analysis, allowing the visualization of collections of frequent itemsets of arbitrary cardinalities. This work also represents a first step towards the unification of clustering and association mining.

References

[1] R. Agrawal, T. Imilienski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," in *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data*, May 1993, pp. 207-216.

[2] R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association Rules," in *Proceedings of the 20th International Conference on Very Large Databases*, Santiago, Chile, September 1994, pp. 487-499.

[3] C. Chen, L. Carr, "Trailblazing the Literature of Hypertext: An author co- citation analysis (1989-1998)," in *Proceedings of the 10th ACM Conference on Hypertext (Hypertext '99)*, Darmstadt, Germany, February, 1999, pp. 51- 60.

[4] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, 1992.

[5] T. Fruchterman, E. Reingold, "Graph Drawing by Force-Directed Placement," *Software—Practice and Experience*, 21, pp. 1129-1164, 1991.

[6] E. Garfield, M. Malin, H. Small, "Citation Data as Science Indicators," in *Toward a Metric of Science: The Advent of Science Indicators*, Y. Elkana, J. Lederberg, R. Merten, A. Thackray, H. Zuckerman (eds.), John Wiley & Sons, New York, 1978, pp. 179-207.

[7] I. Herman, G. Melanon, M. Marshall, "Graph Visualization and Navigation in Information Visualization: a Survey," *IEEE Transactions on Visualization and Computer Graphics*, 6(1), pp. 24-43, 2000.

[8] J. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," in *Proceedings of the ACMSIAM Symposium on Discrete Algorithms*, January 1998, pp. 668-677.

[9] S. Noel, H. Szu, "Multiple-Resolution Clustering for Recursive Divide and Conquer," in *Proceedings of Wavelet Applications IV*, Orlando, FL, April 1997, pp. 266-279.

[10] S. Noel, *Data Mining and Visualization of Reference Associations: Higher Order Citation Analysis*, Ph.D. Dissertation, Center for Advanced Computer Studies, The University of Louisiana at Lafayette, 2000.

[11] S. Noel, V. Raghavan, C.-H. H. Chu, "Visualizing Association Mining Results through Hierarchical Clusters," in *Proc. First IEEE International Conference on Data Mining*, San Jose, California, November 2001, pp. 425- 432.

[12] L. Page, S. Brin, R. Motwani, T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, Stanford Digital Library, Working Paper 1999- 0120, 1998.

[13] H. Small, "Co-citation in the Scientific Literature: A New Measure of the Relationship Between Two Documents," *Journal of the American Society of Information Science*, 24, pp. 265-269, 1973.

[14] H. Small, "Macro-Level Changes in the Structure of Co-Citation Clusters: 1983-1989," *Scientometrics*, 26, pp. 5-20, 1993.

[15] G. Strang, T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge, Wellesley, Massachusetts, 1996.

[16] W. Venables, B. Ripley, *Modern Applied Statistics with S-Plus*, Springer- Verlag, 1994.

[17] J. Wise, J. Thomas, K. Pennock, D. Lantrip, M. Pottier, A. Schur, V. Crow, "Visualizing the Non-Visual: Spatial Analysis and Interaction with Information from Text Documents," in *Proceedings of Information Visualization '95 Symposium*, Atlanta, GA, 1995, pp. 51-58.

[18] The ISI Science Citation Index (SCI), available through the ISI Web of Science, http://www.isinet.com/isi/products/citation/sci/.

[19] R. Baeza-Yates, B. Ribeiro-Neto (eds.), *Modern Information Retrieval*, Addison Wesley Longman, 1999.