DATA MINING AND VISUALIZATION OF REFERENCE ASSOCIATIONS: HIGHER ORDER CITATION ANALYSIS

A Dissertation

Presented to the

Graduate Faculty of the

University of Louisiana at Lafayette

In Partial Fulfillment of the

Requirements for the Degree

Doctor of Philosophy

Steven Earl Noel

Fall 2000

DATA MINING AND VISUALIZATION OF REFERENCE ASSOCIATIONS: HIGHER ORDER CITATION ANALYSIS

Steven E. Noel

APPROVED:

Vijay Raghavan, Co-Chair Distinguished Professor in Computer Science	Chee-Hung Henry Chu, Co-Chair Associate Professor of Computer Engineering
Miroslav Kubat Associate Professor of Computer Science	Lewis Pyenson Dean, Graduate School

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In some respect, the World Wide Web is like a vast library without an index system. Search engines are thus critical in finding Web pages of interest. Traditionally, search engines rank their results according to how well pages match keywords in the user query. In contrast, more innovative search engines such as Google [Henz00] first perform a keyword search, and then analyze the structure of Web hyperlinks to generate page ranks, independent of user queries for the selected pages. However, the results for these link-based search engines are still displayed as ranked lists, just as for traditional search engines.

Simple linear lists cannot adequately capture many of the complex hyperlink relationships among Web pages. Techniques from the field of information visualization [Tuft91][Card99] can help in this regard, making complex relationships more readily understandable. Visualization augments serial language processing with eye/brain parallel processing. Thus, the goal of visualization techniques is to enable users to recognize patterns in Web link structure, in turn helping to alleviate cyberspace information overload.

Previous approaches in this area have typically analyzed Web hyperlinks directly to determine page relationships [Klei98], or have relied on measures of similarity that only consider joint referencing of pairs of pages. The approach proposed in this work relies instead on measures of similarity among sets of pages of arbitrary cardinality. In

particular, the similarity among a set of pages is based on the number of other pages that jointly link to them.

The proposed similarity measures are inspired by the concept of co-citations, introduced in classical information retrieval in the context of citations appearing in published literature [Whit89]. Co-citations reduce complex citation or hyperlink graphs to simple scalar similarities between documents or Web pages. Co-citation based similarities allow the direct application of standard tools developed in other areas of science, such as cluster analysis [Vena94] and the minimum spanning tree [Corm96].

Similarity among objects by common reference has recently received some attention in the form of association mining [Agra93], which is a sub-field of data mining. While they are not usually recognized as such, what are defined as itemsets in association mining can be interpreted as generalized co-citations. Similarities between pairs of documents in co-citation analysis can be generalized to reflect the impact of sets of documents of arbitrary, larger cardinality that are jointly cited. Thus, itemsets are interpreted as higher-order co-citations.

This work is the first known application of itemsets to the visualization of link structures. Important (frequently occurring) higher-order itemsets are often obscured by the mere pairwise treatment of traditional co-citation analysis [Smal73]. The approach I take here involves the discovery of frequently occurring itemsets of arbitrary cardinalities, and the assigning of importance to them according to their frequencies. The generalization of co-citations to itemsets also enables user-oriented clustering [Bhuy91a][Bhuy91b][Bhuy97], where the user is allowed to specify weight of importance to larger sets of documents, beyond just pairs.

Because a collection of itemsets is not a disjoint set, there is a combinatorial explosion in the numbers of sets the user has to potentially deal with. I propose a novel approach to the problem of presenting results of association mining to users, which involves embedding higher-order co-citations (itemset supports) into pairwise document similarities. This hybrid of pairwise and higher-order similarities greatly reduces the complexity of user interaction, while being significantly more consistent with higher-order co-citations than standard pairwise similarities. It also admits the application of fast algorithms developed for data mining, which are empirically known to scale linearly with problem size [Agra94].

Mathematically, pairwise similarities can be modeled as a fully connected graph, to which clustering or minimum spanning tree algorithms can be applied. In the case of higher-order similarities, this graph is generalized to a hypergraph, i.e. a graph whose edges span more than just pairs of vertices. My approach of embedding higher-order co-citations in pairwise similarities eliminates the difficult task of forming clusters or minimum spanning trees directly from a hypergraph. Instead, standard algorithms may be directly applied.

The importance of clustering in information retrieval is well known [Baez99]. Link analysis in general provides a broadening of search results, by identifying documents that are linked to the initial set of documents matching the query. Clustering, in addition, can provide a narrowing of search results, by allowing the user to focus on documents in pertinent clusters only, while excluding other documents. In other words, as a result of this work, link analysis can be applied for both broadening and narrowing of search results.

The application of the proposed higher-order similarities to clustering algorithms greatly increases the tendency for important frequently occurring itemsets to appear together in clusters. This tendency is measured by a new metric I introduce specifically for comparing clusters to frequently occurring itemsets.

Moreover, I offer a theoretical guarantee that there is always a sufficient degree of nonlinearity one can apply to itemset supports (frequencies of occurrence) such that more frequent itemsets get placed together in clusters at the expense of less frequent ones. This guarantee relies on asymptotic growth bounds for nonlinearly transformed itemset supports. More specifically, the nonlinearly transformed support of the most frequently occurring itemset asymptotically bounds from above the nonlinearly transformed supports of all other itemsets. This means that distances between documents in the most frequent itemset can all be made smaller than distances to any documents outside that itemset, thus guaranteeing that the most frequent itemset will form a cluster. This argument can be extended to cover all other itemsets, based on their relative supports and overlap of documents.

My method of embedding itemset supports in pairwise similarities is particularly successful when the more frequently occurring itemsets are comparatively sparse. I therefore investigate citation itemset support distributions. That is, I show the frequency of occurrence of co-citations of a given order (itemsets of a given cardinality), for various science citation data sets.

For reasons of computational feasibility with large document collections, citation analysis has traditionally used the single-linkage clustering criterion only [Garf79]. Given the computational power of modern machines, stronger clustering criteria such as

average or complete linkage becomes feasible. I show that in the context of citation databases, single-linkage clustering alone is insufficient for completely characterizing the cluster structure of typical document collections. In particular, clustering results are usually quite different for each of the clustering criteria.

Previous approaches to co-citation based clustering either exclude visualization altogether, or visualize a single clustering corresponding to *a priori* numbers of clusters or single threshold similarity [Garf79][Smal93]. Instead, I apply the dendrogram visualization [Vena94], which shows the hierarchy of clusters for all possible thresholds, with no *a priori* requirement for the desired number of clusters. This is the first time that the dendrogram has been proposed for the visualization of either hypertext systems or document citation databases.

I introduce the concept of an "augmented dendrogram" for the visualization of significant (document) item associations. The augmented dendrogram highlights items that are a part of the same itemset, via graphical glyphs. This extension of the standard dendrogram allows the simultaneous visualization of both hierarchical clusters and important higher-cardinality itemsets.

The feasibility of the augmented dendrogram depends on a sufficiently small number of highlighted itemsets having items in common. When an item appears in too many highlighted itemsets, the augmented dendrogram becomes unwieldy. At this point one must rely on non-augmented dendrograms computed from the new hybrid pairwise/higher-order distances.

The dendrogram augmentation also includes the addition of textual information for documents being clustered. The leaves of the dendrogram tree correspond to

individual documents. The augmented dendrogram adds document bibliographic details to each leaf, thus supporting information retrieval.

The minimum spanning tree has been shown to provide a network of literature influences among collections of documents [Chen99a][Chen99b]. In this dissertation, I apply my new higher-order document similarities to minimum spanning tree visualizations. In particular, I investigate the effects that higher-order distance functions have on the influences of documents that are members of frequently occurring itemsets.

I propose three new metrics for measuring the effects of distances on frequent itemset members within the minimum spanning tree influence network. The first metric measures the connectedness of itemset members in the network. This is for testing the hypothesis that hybrid pairwise/higher-order distances increase the connectedness of members of frequently occurring itemsets. The other two metrics measure, respectively, the direct and total influences of an itemset member. They help test the hypothesis that the new hybrid distances increase the influence that members of frequently occurring itemsets have within the network.

I also introduce a novel method for the landscape visualization of a minimum spanning tree's node density, based on the wavelet transform. This visualization is considered "2.5-dimensional," being a two-dimensional landscape surface embedded in three dimensions. The landscape surface offers depth cues to help users recognize node positions. It also helps to alleviate the disorientation that often occurs with three-dimensional visualizations, since humans are adept at navigating landscapes. For this visualization I apply a force-directed layout algorithm for positioning nodes of the minimum spanning tree [Fruc91].

I introduce a novel approach to clustering based on the landscape visualization of the minimum spanning tree. The visualization is modified to show clusters by retaining only the tree edges between documents of the cluster. For example, single-linkage clusters are visualized by removing minimum spanning tree edges larger than some threshold amount.

Unlike the single-linkage approach, which is applied to the original edge distances, I propose the application of the threshold to the edge distances induced by the force-directed layout algorithm. The result is a new type of clustering in which clusters are oriented to highly influential documents, and highly influential documents themselves are placed in separate clusters. This is in contrast to traditional clustering methods, in which highly influential documents are placed *together* in clusters by virtue of the relatively small distances between them.

Interestingly, such clusters correspond approximately to connected components of the wavelet density landscape after the application of a threshold. Changes to the threshold value result in a nesting of connected components, which corresponds to a clustering hierarchy. Overall, I interpret the new wavelet landscape visualization as a form of spatial, hierarchical, fuzzy clustering.

I introduce the novel “augmented minimum spanning tree” for visualizing significant document associations. This extension of the standard minimum spanning tree visualization highlights documents that are part of the same itemset, allowing them to be readily identified within the tree. Like the augmented dendrogram, the augmented minimum spanning tree includes text for document nodes, as an aid to information retrieval.

The proposed methods of data mining and visualization are evaluated using data sets extracted from the Institute for Scientific Information's (ISI) Science Citation Index (SCI). The SCI is a component of ISI's Web of Science [WOS00], which provides access to citation databases that cover over 8,000 international journals. The application of data mining and visualization to science citations is consistent with the interests of this work's sponsor, the U. S. Department of Energy's Office of Scientific and Technical Information (DoE OSTI) [OSTI00].

But more generally, my approach is applicable to any information space in which objects may be associated by reference, particularly spaces modeled by directed graphs. Examples abound in such areas as software engineering, market analysis, communications networks, and perhaps most notably the World Wide Web.

The next chapter reviews previous approaches, and provides the background and further motivation for this work. It first describes literature citation analysis in the area of bibliometrics. It then covers analyses of link structure for information retrieval and visualization, which often rely on results from classical citation analysis. Next it describes association mining, including fast algorithms for computing frequently occurring itemsets. It then shows how advances in information visualization can contribute to comprehension of some potentially complex relationships among linked objects.

Chapter 3 introduces itemset supports as indicators of higher-order co-citation similarities, and describes my proposal for embedding them into pairwise similarities for clustering visualizations. It begins with some foundational issues in co-citation analysis, including the conversion of similarities to dissimilarities (distances) to facilitate the

application of clustering algorithms. It then describes hierarchical topological clustering, in particular single-linkage, average-linkage, and complete-linkage clustering, and describes the dendrogram visualization of cluster hierarchies.

Chapter 3 also introduces a metric that compares a given clustering to a set of significant itemsets, e.g. ones that occur frequently. The metric helps guide the design of the new inter-document distances that include higher-order co-citation similarities, i.e. hybrid pairwise/higher-order distances. The metric is then applied in a number of computational experiments with literature citation data sets, to test my proposed approach to document clustering.

Chapter 4 investigates methods of reducing the computational complexity of inter-document distances. It first applies fast algorithms for computing more frequently occurring itemsets in hybrid distances. It then proposes a model for itemset support distributions, the rapid decay of the distributions for larger supports providing additional evidence that fast algorithms for computing frequent itemsets scale linearly with problem size. Chapter 4 also offers and some experimental evidence that simple schemes for weighting of transactions or documents in computing pairwise distances is insufficient for consistency between clusters and frequent itemsets.

Chapter 5 covers the application of higher-order co-citations to the minimum spanning tree visualization. It first introduces the minimum spanning tree problem and algorithms for solving it. Next it describes the force-directed algorithm for positioning nodes of the minimum spanning tree. It then proposes three separate itemset-based evaluations of the minimum spanning tree: a metric for average number of connected components on the tree formed by an itemset, a metric for average vertex degree of an

itemset member, and a metric for the numbers of tree descendents of itemset members. Finally, Chapter 5 describes the minimum spanning tree landscape visualization and its interpretation as a novel approach for clustering.

Chapter 6 summarizes this dissertation, and highlights its conclusions. It also suggests ideas for future work in this area, including higher-order similarities for user-oriented clustering, inferring association rules from hierarchical clusters, applying maximal frequent itemsets (frequent itemsets that are not subsets of other ones), and extending the minimum spanning tree visualization to three dimensions.

Chapter 2

Background, Motivation, and Previous Work

This chapter provides the background and motivation for the approach to visual-based information retrieval taken in this dissertation. It also describes previous work that pertains to my approach. The material covered in this chapter falls into 4 general categories: citation analysis, link analysis, association mining, and information visualization.

A typical example of visualization-based analysis of link structures is the analysis of citations in published works. This early work has lead to the important co-citation relationship between documents, along with methods of clustering and visualizing document collections. Citation analysis is described in Section 2.1.

With the advent of the World Wide Web, much attention has been paid to link analysis. This work has resulted in more sophisticated forms of link analysis, including methods of detecting and measuring various link structures. Advances in link analysis in this context are covered in Section 2.2.

In later chapters, I will propose a novel type of analysis that is applicable to link structures. This analysis is based on association mining of items that co-occur in transactions, taken from the field of data mining. I describe the task of association mining for itemsets and fast algorithms for computing them in Section 2.3.

Methods of hypertext analysis yield structures and relationships that are generally quite complex. The level of complexity can be overwhelming when these methods are

not supported with visualization strategies. There has been a substantial amount of work in visual-based methods for information understanding. Developments in information visualization that pertain to this dissertation are described in Section 2.4.

2.1 Citation Analysis

Researchers in science and technology expend a great deal of labor in understanding the complex relationships among documents in their fields of study. This understanding is also critical for science and technology program managers. For example, they may need to see the impact of a funded project on the scientific community, to detect emerging technologies, to determine the maturity of a field, or to access the cross-disciplinary nature of a work. Achieving such an understanding of the literature is time consuming and labor intensive. Moreover, managers are often responsible for projects in multiple areas of science, many of which are not their areas of expertise. The needs of science and technology program managers are of particular importance to our research sponsor [OSTI00].

While it is often underutilized, much can be learned through the analysis of document citations. Citation analysis was pioneered in the late 1950's, by Eugene Garfield [Garf79]. One of Garfield's important early contributions is the "historiograph," a visualization of the graph structure of document citations. Figure 2-1 shows early examples of historiographs.

In the early 1970s, Small introduced the idea of co-citations [Sma73]. A co-citation is an association between 2 documents, meaning that they are both cited by some other document. The number of co-citations for a given pair of documents serves as a

measure of similarity between them. The application of co-citation document similarities reduces the complex graph of citations among documents to a simpler statistical summary, so that co-citation similarity serves as a compact representation of citation graph topology.

Garfield later applied co-citation similarities for document cluster analysis. That is, he computed disjoint sets (or clusters) of documents such that documents within clusters are more similar to one another than to documents outside clusters, with similarity being based on co-citation counts. He then visualized clusters via "cluster maps," which show documents within a cluster, with links between documents that are deemed sufficiently similar. Figure 2-2 shows an example cluster map.

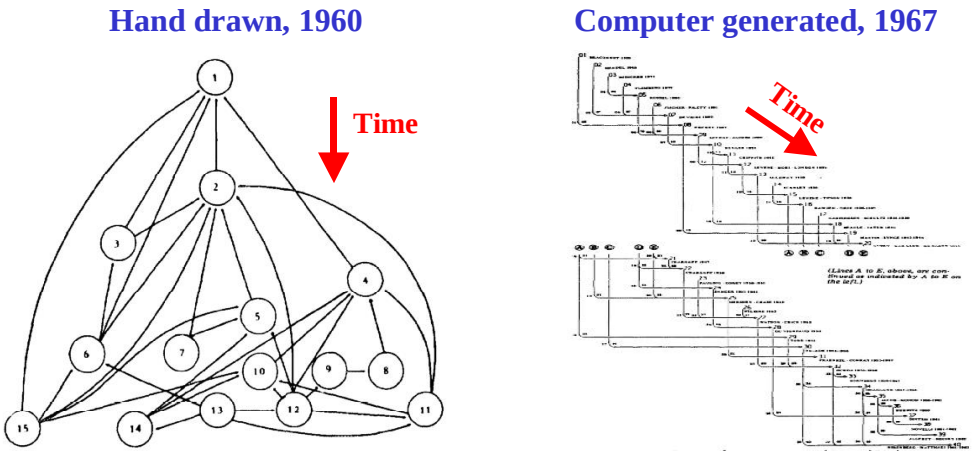


Figure 2-1: Garfield's historiographs are a pioneering visualization of citation link structures.

Garfield's citation-based visualizations can show, for example, the historical evolution or intellectual structure of a discipline, core clusters within a discipline, or relatedness among disciplines. Patterns within the structure of document citations may

also mirror larger events in society. Analysis tools supporting such visualizations include hierarchical clustering, multidimensional scaling, and factor analysis.

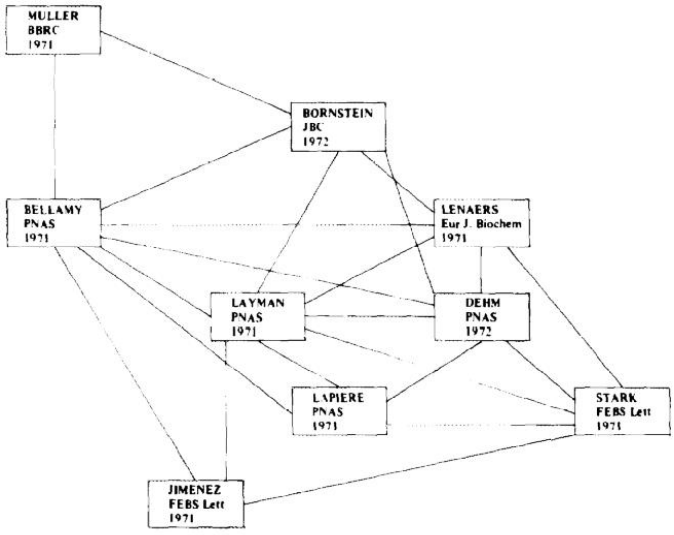


Figure 2-2: Garfield's cluster maps for visualizing co-citation based clusters of documents.

It can be difficult to anticipate the exact questions program managers or other workers may try to answer through the analysis of document citation structure. However, it is possible to elaborate on some questions that are likely to be of general concern. These questions can then guide the design of a general framework for information retrieval based on citation analysis.

An example problem that may be of general concern is the ability to assess the relative maturity of an area of science or technology. This may be deduced from the historiograph, a time-ordered graph showing the structure of document citations, and thereby the historical evolution of a discipline or set of disciplines.

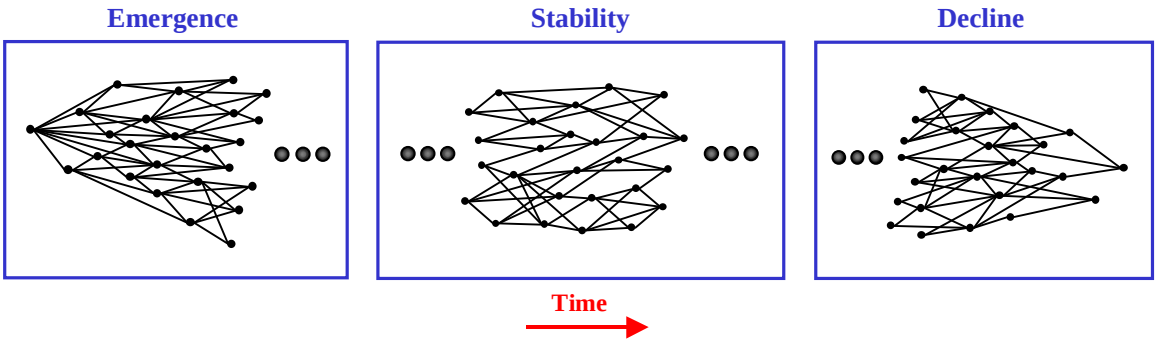


Figure 2-3: Typical document citation structures for emerging, stable, and declining disciplines.

Figure 2-3 shows typical historiographs for the cases of emerging, stable, and declining areas of study, respectively. The nodes of the historiograph are documents. Their horizontal positions are determined by time of publication, while their vertical positions are arbitrary. The edges of the graph are citations among documents. While the graph is actually directed, edge directions can be inferred from the graph layout. That is, time imposes an ordering of document nodes.

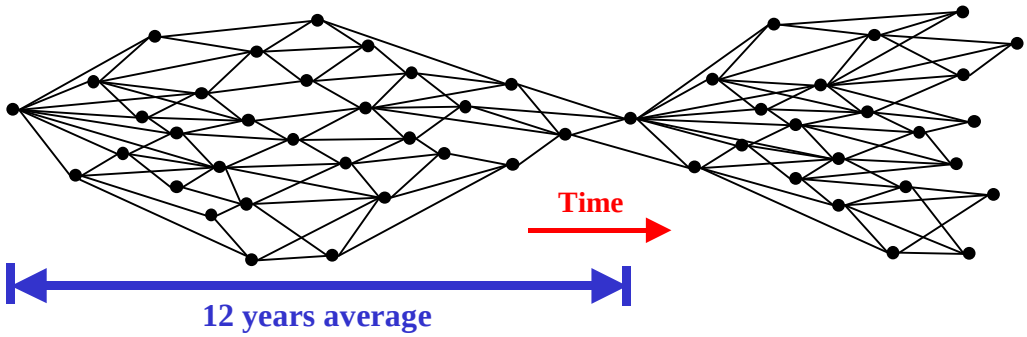


Figure 2-4: Citation structure of a discipline under the oscillating model.

There is also evidence that some disciplines exhibit structural oscillations between expansion and contraction [Smal93], as shown in Figure 2-4. It has been hypothesized

that this corresponds to alternating periods of discovery and consolidation within the discipline.

The idea is that disciplines tend to follow a general line of development or life cycle, beginning with a discovery that brings about a revolutionary paradigm shift. This shift is then followed by a rapid expansion of papers exploiting the discovery. Eventually a stable middle period is reached, which may be characterized for example by the appearance of review papers or a shift to applied science or technology. At some point the discipline experiences decline, perhaps coinciding with a period of internal reflection or rebuilding. Apparently in this later part of the life cycle new revolutions are more likely to occur, in which the cycle is restarted. This oscillation of disciplines is consistent with the idea of punctuated equilibrium.

A central problem in citation analysis is to be able to determine clusters within the structure of document citations. Clusters based on co-citations are known to generally correspond well with individual disciplines [Garf78]. Analysis of document clusters and their interaction over time yields insight into the evolution of science, both within and among disciplines. Such analysis includes hierarchical clustering for determining cohesive document groups at various discipline scales, and spatial visualization of clusters via multidimensional scaling.

The proliferation of electronic publishing has increased the rate of published documents, making the program manager's job even more difficult. There is also a large and growing body of unpublished "gray" literature, such as scientific databases and web documents. Fortunately, electronic documents also provide the opportunity for computerized processing. In the area of citation analysis, there exist databases of

scientific document citations. These are exemplified by the Web of Science [WOS00], developed at the Institute for Scientific Information under Garfield's guidance.

The standard approach in citation analysis is to cluster documents within a collection, corresponding to various scientific fields. Figure 2-5 shows an example of such citation-based clustering. This type of clustering typically uses the frequency of co-citation as a document similarity measure. It considers documents in pairs, as is done for traditional graph-theoretic clustering algorithms in general.

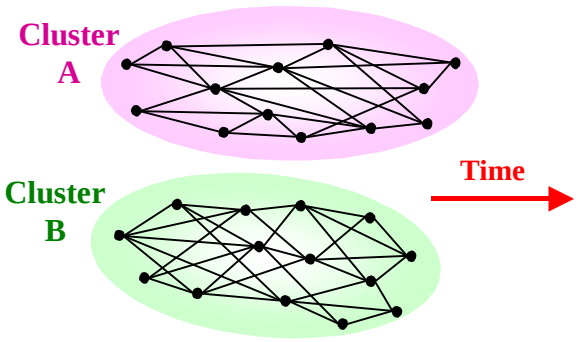


Figure 2-5: Clustering of documents based on citation structure.

Moreover, traditional co-citation analysis relies on simple single-linkage clustering, because of its lower computational complexity given the typically large number of documents in a collection. Researchers concede a weakness in this single-linkage clustering approach [Smal93]. The concern is with the possible "chaining" effect, in which unrelated documents get clustered together through a chain of intermediate documents. Figure 2-6 shows an example of single-linkage chaining, in which 2 clusters merge by single co-citation link.

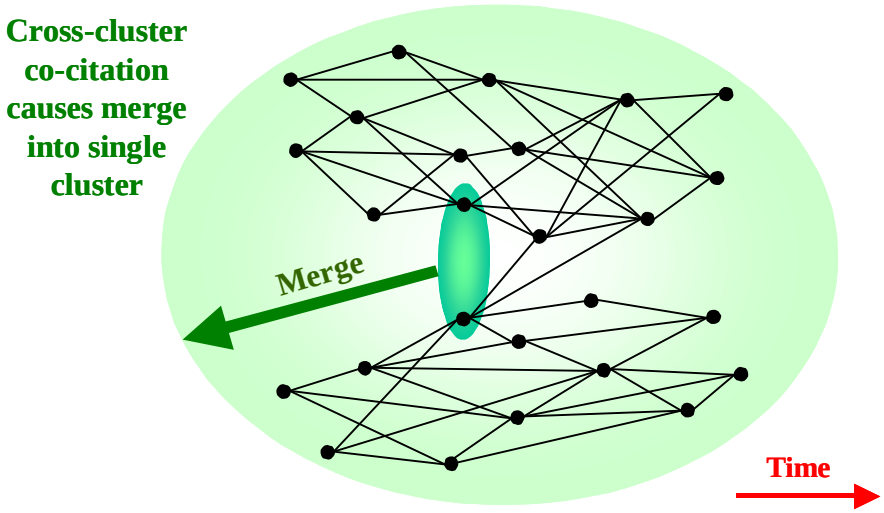


Figure 2-6: Example of chaining in co-citation single-linkage clustering.

Given the improved processing speeds of computing machines, it becomes feasible to apply stronger clustering criteria in citation analysis. In later chapters, I propose a novel clustering approach involving higher-order similarities among documents. That is, similarities for document pairs are generalized to include similarities among document triples, sets of 4, and so on, up to the full number of documents in the collection. In other words, co-citation of a document pair is generalized to 3rd-order citation, 4th-order citation, etc. This higher-order similarity admits a stronger clustering criterion in which a sufficient number of other documents must *simultaneously* cite a given set of documents for them to be considered a cluster.

This new criterion is certainly stronger than single-linkage clustering, in which a document need only be sufficiently co-cited with a *single* member of a cluster to be included in that cluster. It is an even stronger criterion than complete-linkage clustering,

the strongest of the traditional graph-theoretic clustering methods, in which a document must be sufficiently co-cited with *all* other cluster members to be included in the cluster. The key distinction is that here only document *pairs* are considered in the various co-citations, versus the higher-order sets in our approach.

Such a strong clustering criterion insures that documents within a cluster are *directly* rather than *indirectly* related to one another. The resulting smaller, more cohesive clusters should better support micro-scale studies of document collections. Documents within these stronger clusters would also be more relevant to one another for the purpose of information retrieval.

The stronger criterion allows a document to be simultaneously a member of more than one cluster. We do not have clusters in the traditional sense of disjoint sets, but rather sets of documents that have been frequently cited together. These potentially overlapping sets may better reflect the actual thematic content of the documents. However, their overlap leads to combinatorially exploding numbers of sets, which can disorient the user. In later chapters, I propose an approach for overcoming this.

2.2 Link Analysis

Early research in information retrieval focused on word-based techniques for finding documents that are relevant to user queries. Some early work also analyzed link structures of documents, usually collections of documents linked by citation. But the proliferation of the World Wide Web has resulted in large numbers of hyperlinked documents, created by independent authors. This has lead to the formation of the area known as link analysis within information retrieval.

A Web hyperlink is a reference from one Web page to another, in which the selection of the reference on the referring page invokes the referred page. While hyperlinks are sometimes mere navigational aids (such as providing quick access to the Web site's home page), they generally refer to pages whose content supports that of the referring page. Methods of link analysis thus assume that if pages are linked, they are more like to have similar content.

One of the better known applications of link analysis is the ordering of documents resulting from a user query. There are two general approaches to link-based page ranking: query-independent ranking and query-dependent ranking. Query-independent ranking attempts to measure the intrinsic quality of Web pages, independent of a particular user query. Query-dependent ranking measures the qualities of pages in terms of their relevance to a given user query.

A popular query-independent measure of page rank is PageRank, developed by Brin and Page [Page98]. It applies the number of hyperlinks referring to a page as a quality criterion, but weights each hyperlink according to the quality of the referring page.

PageRank is computed recursively, via

$$R(v) = \frac{d}{n} + (1-d) \sum_{(u,v) \in G} \left[\frac{R(u)}{\text{outdegree}(u)} \right], \tag{2.1}$$

where $R(v)$ is the PageRank for page v. Here, the Web is modeled as a graph G containing a vertex for each page u, and a directed edge (u, v) if page u links to page v. The number of vertices in G is n, and $\text{outdegree}(u)$ is the number of graph edges

leaving page u. The parameter d is a dampening factor, usually set between 0.1 and 0.2.

PageRank is known to generally work well in distinguishing between high and low quality Web pages. Although no theoretical bounds are known for the number of iterations necessary for convergence, in practice less than 100 iterations usually suffice.

A popular algorithm for computing query-dependent page rank is HITS, developed by Kleinberg [Klei98]. The algorithm first generates a Web sub-graph known as the *neighborhood graph*, generated from an initial set of query-matching documents and those documents that are linked (in either direction) to the initial set. Documents in the neighborhood set are then ranked by *hub* scores and *authority* scores. Documents with high authority scores should have relevant content, while those with high hub scores should point to documents with relevant content.

Hub and authority scores are computed recursively, as

$$A(v) = \sum_{(u,v) \in G} H(u) \tag{2.2}$$

and

$$H(v) = \sum_{(v,u) \in G} H(v). \tag{2.3}$$

Here, for page v, $A(v)$ is the authority score and $H(v)$ is the hub score, where (u, v) is a directed edge in the neighborhood graph G. The intuition begins with the idea that pages with large in-degree might be good authorities, and pages with large out-degree might be good hubs. Then under recursion, pages that are pointed to by many good hubs might be even better authorities, and pages that point to many good authorities might be even better hubs.

Like for PageRank, the computation of hub and authority scores has no known bound on the number of iterations. But in practice, the scores generally converge quickly. The HITS algorithm does potentially suffer from a problem known as "topic drift." If the neighborhood graph contains pages that are not relevant to the query (via links to initial query-matching pages), they could be given high scores, despite the fact that they are irrelevant to the search topic.

Terveen *et al* define a link structure that groups together sets of related sites, known as the clan graph [Terv98]. For an *n*-clan graph, each node is connected to every other node by a path of length *n* or less, and all connecting paths go only through nodes in the clan. Construction of the clan graph tends to filter out irrelevant sites, and discovers additional relevant ones. It is reported to be robust with respect to "noise" in the form of off-topic pages in the initial set.

Mukherjea *et al* describe a link structure called pre-trees, and apply the structure to visualization of the Web [Mukh94a][Mukh94b]. Pre-trees are generalized trees in which there is a root node, but children need not be trees – they can be arbitrary graphs (called branches). Branches are restricted to having no links between them. While not originally called such, the pre-tree data structure has traditionally been applied in top-down clustering [Hart75].

Mukherjea *et al* apply pre-trees to induce hierarchies in Web graphs, then visualize the resulting hierarchies. Their algorithm identifies potential pre-trees through a combination of content and structural analyses. It then ranks the pre -tree via a metric that includes measures of information lost in forming the pre-tree, how well pre-tree branches approximate actual trees, and how well the choice of root yields a shallow tree.

2.3 Association Mining

In subsequent chapters, I propose a generalization of co-citations that extends them from a relationship between 2 documents to a relationship among any number of documents. Interestingly, this higher-order extension is equivalent to "itemsets supports" in the field of data mining [Rysz98]. In data mining, methods of machine learning are applied to databases to discover novel, interesting, and sometimes surprising pieces of knowledge. Data mining is best known for its financial applications, such as marketing, investing, and risk assessment for credit or insurance.

An important part of data mining is forming association rules among items in the database. Associations are computed directly from itemset supports, which depend on the simultaneous occurrence of items within database transactions. Association mining is perhaps best known for its application to market purchases, the so-called "market basket" problem. The associations are made among market items based on how frequently they are purchased together.

An analogy can be made between market purchases and document citations. The market items are analogous to documents, and the purchase of items is analogous to the citation of documents. Thus association mining is applicable to citation analysis, forming associations among groups of documents that are frequently cited together. This mining would enhance understanding of the structure of document citations, making explicit those associations that may otherwise go unnoticed. Additional insight could be gained by forming citation associations at levels other than individual documents, for example authors, publications, institutions, or countries.

The analogy between market purchases and document citations is not perfect. In the former, the entities making purchases (people) are distinct from the entities they purchase (items), whereas in the latter there is no such distinction (they are all documents). In the case of citations, there thus exists certain kind of symmetry in which clustering may be applied to either cit*ing* or cit*ed* documents. This is equivalent to the symmetry between co-citations and "bibliographic coupling" within traditional citation analysis [Garf79].

An argument has been made against the application of association mining for dynamic environments such as the World Wide Web. The concern is that to insure updated results, the mining would need to be completely redone whenever the data items change. A data structure has been proposed to address association mining in dynamic environments, called the itemset tree [Hafe99].

Rather than re-computing itemset supports after database changes, supports could be stored, then updated with the new information. This trades time complexity for space complexity. However, the lattice of itemset supports potentially grows exponentially with respect to numbers of item sets. The itemset tree provides for the storage of itemset supports, but has lower space complexity than the itemset lattice. The structure eliminates the need for accessing the full database when new transactions (e.g. document citations) are added, though with some overhead for inserting new transactions.

An important part of association mining is the computation of all itemset supports that exceed a given threshold, i.e. the frequent-itemset problem. This is a critical problem, since association rules are computed directly from supports, and more frequent itemsets represent those with stronger associations.

The frequent-itemset problem is generally considered to be the computational bottleneck in association mining, because of the complexity of the itemset lattice. Efficient algorithms have been proposed than retain the exponential worst-case complexity [Agra94]. However, they have been shown empirically to scale linearly with problem size. New classes of document distances that I propose in later chapters are able to take advantage of these fast algorithms for computing frequent itemsets.

2.4 Information Visualization

Because of the complexity of document citations and their associations, visualization becomes critical. A traditional approach, developed by Garfield [Garf79], is to embed documents or document clusters as points in a plane. The embedding algorithms attempt to preserve the original citation-based similarities among documents, which are interpreted as Euclidean distances in the plane. The resulting display for a given cluster of documents is a "cluster map," reminiscent of a 2-dimensional scatterplot.

My new approaches to document visualization maintain this embedding of documents or document clusters within a plane, at least as an initial step. I point out that this is merely a mechanism for inducing spatial coordinates upon (or "spatializing") inherently non-spatial data. Such spatialization is common in the emerging field of information visualization [Card99], which was pioneered by Edward Tufte [Tuft91].

A recent development in citation analysis is to visualize the minimum spanning tree among documents [Chen99a][Chen99b]. The tree shows the minimal set of essential links among a set of documents, with respect to co-citation based distances. Spatialization of the minimum spanning is accomplished through a heuristic that models

repulsive forces among tree vertices and springs for tree edges, known as a force-directed approach.

The minimum spanning tree is interpreted as a network of influences among the documents, with highly influential core documents in the center and the emerging research front on the edges. Branches in the network correspond to bifurcations of ideas in the evolution of science. Chen's minimum spanning tree network of influence is shown in Figure 2-7. Figure 2-8 shows Chen's StarWalker VRML application for Web-based navigation of the influence network.

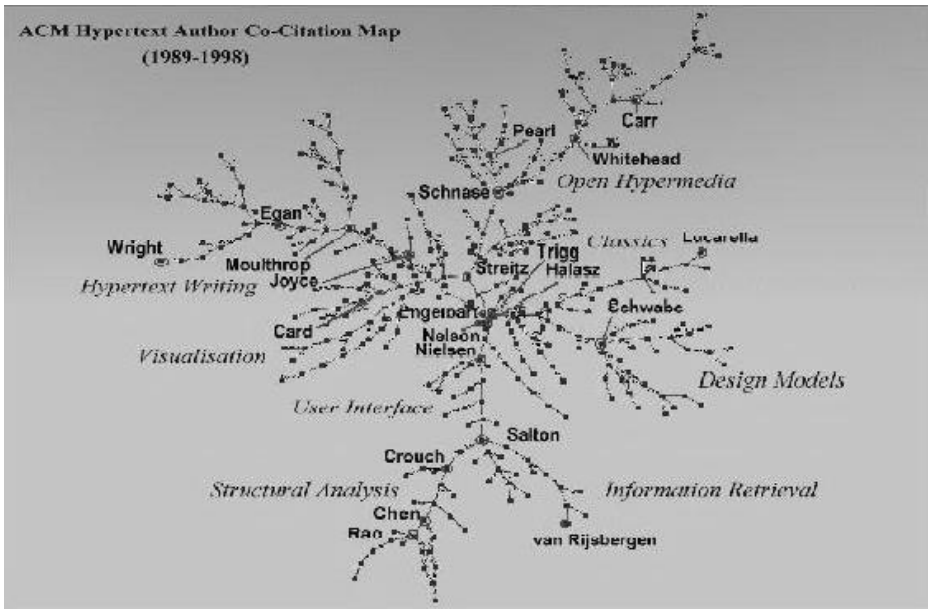


Figure 2-7: Chen's visualization of author influence network.

In my work, I retain the visualization of minimum spanning tree, including a force-directed approach for spatialization. But I extend the minimum spanning tree visualization in a number of ways. For example, I propose showing clusters explicitly in the visualization, by retaining only tree edges among cluster members. When this is

based on the application of a threshold distance to edges induced by the force-directed spatialization algorithm, the result is a new type of clustering. Clusters are oriented to highly influential documents, and highly influential documents themselves are placed in separate clusters. I also extend the visualization by highlighting association mining document itemsets (for identifying significant document associations) and including text for document nodes (as an aid to information retrieval).

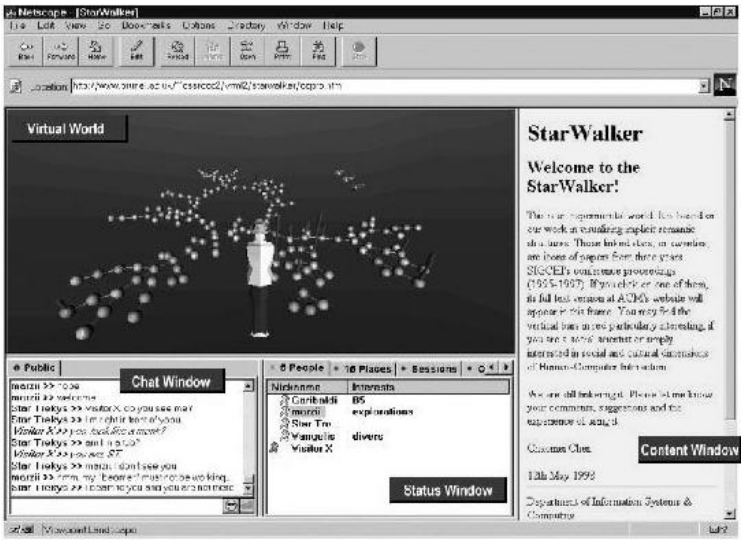


Figure 2-8: Chen's StarWalker VRML application for navigation of influence network.

Another extension I propose for the minimum spanning tree is inspired by the "ThemeScape" visualization developed at Pacific Northwest Laboratory [Wise95], shown in Figure 2-9. This method of information visualization estimates the density of document points in a scatter plot, where distances are obtained from text analysis. The visualization resembles a landscape, in which peaks correspond to clusters of documents, and valleys correspond to the distances between them. Visualizing the document

landscape helps one understand the density structure of a collection directly, rather than having to infer it from the document points.

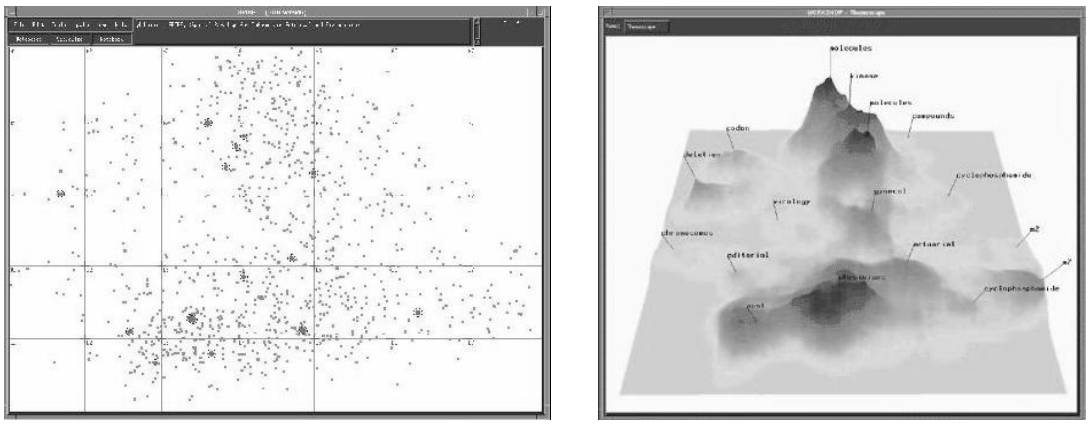


Figure 2-9: Visualizations of document collections from Pacific Northwest Laboratory: Galaxies scatter plot and ThemeScapes document landscape.

While the ThemeScape visualization has been applied to text analysis, I propose for the first time its application to citation analysis. In particular, I apply it as a visualization of the minimum spanning tree, with distances computed from co-citations. I extend the standard ThemeScape visualization by embedding the tree vertices and edges within the density landscape surface. This allows a direct three-dimensional interaction with the document points, while the landscape surface provides depth cues that alleviate the disorientation typical of visualizing points in three dimensions.

Moreover, I extend the ThemeScape so that the landscape density surface can be visualized at a variety of spatial resolutions [Noel97]. At lower resolutions, the landscape features are larger and smoother, corresponding to higher-level clusters. At higher resolutions, the larger features resolve into smaller ones, corresponding to lower-

level clusters. This is analogous to hierarchical clustering, where the level of resolution is analogous to the clustering threshold.

Hendley *et al* have proposed the visualization of Web pages in three dimensions [Hend95]. That is, coordinates are induced for the vertices corresponding to individual pages, through an iterative process that simulates a physical system of springs corresponding to graph edges. Here the edges represent actual Web hyperlinks. In contrast, my approach is to visualize the minimum spanning tree resulting from co-citation similarities based on the actual hyperlinks. The Narcissus visualization is shown in Figure 2-10.

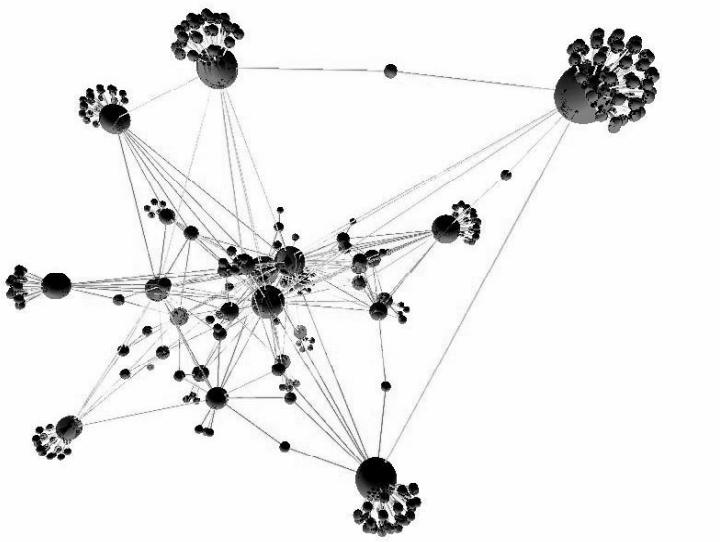


Figure 2-10: Narcissus visualizes Web links directly, rather than employing measures of page similarity derived from link information.

Utilizing a 3rd spatial dimension improves the ability to generate vertex coordinates under a force-directed graph layout algorithm, since it is easier for vertices to

move pass one another. However, such three-dimensional visualizations are known to potentially disorient the user, particularly for graph visualizations.

An important idea in modern information visualization is known as "Focus+Context." The idea is that the user should be able to focus attention on a portion of the visualized space, with the remaining space being de-emphasized but included as context. This is often accomplished by a nonlinear conformal mapping of the space being visualized, with under interactive control. The conformal warping is designed so that a specific portion of the space is exaggerated, to provide focus.

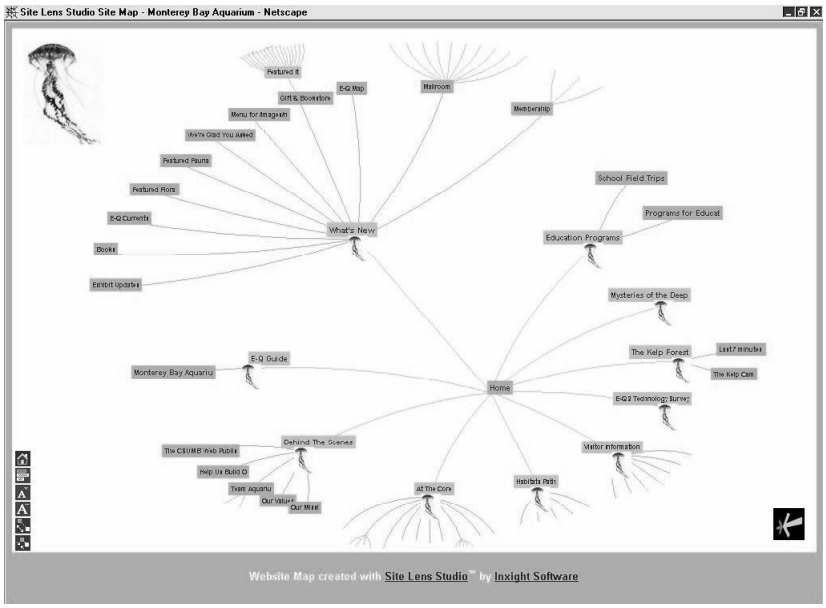


Figure 2-11: InXight's hyperbolic lens visualization for Focus+Context.

One of the better-known examples of Focus+Context distortion is Site Lens, developed at Inxight. Site Lens employs a mapping from a rectangular to hyperbolic space [Lamp95], which inherently has a focus point. Site Lens is shown in Figure 2-11. Munzner has proposed a three-dimensional generalization of the hyperbolic lens

[Munz95][Munz97], shown in Figure 2-12. Such Focus+context techniques are complementary to the approach I take in this dissertation, in that they could be applied to any of my proposed visualizations.

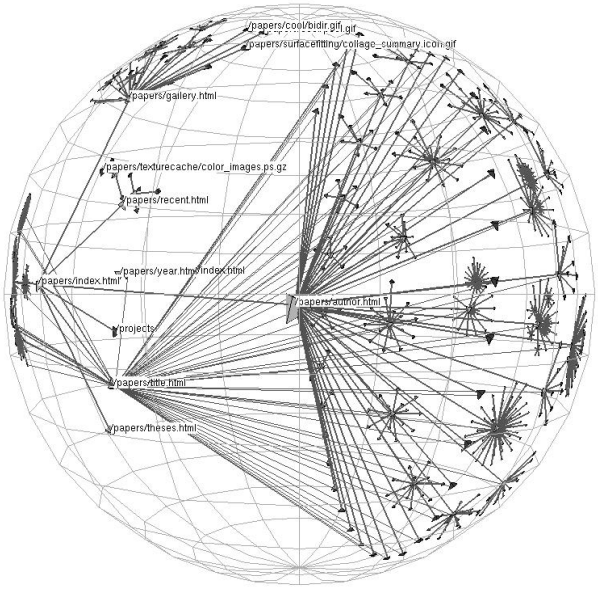


Figure 2-12: Munzner's 3-dimensional generalization of the hyperbolic lens visualization.

Another important idea in information visualization is the *dynamic view*, in which the display changes in response to user action, for example by movement, distortion, or culling. An example of dynamic view that is particularly applicable to the work in this dissertation is the cone tree visualization [Robe91], developed at Xerox PARC (Palo Alto Research Center). The cone tree is a 3-dimensional tree for visualizing hierarchies, as shown in Figure 2-13. The tree rotates in order to bring forward selected nodes, thereby providing focus. The tree can also be culled either automatically or manually, to include only tree subsets of interest. The cone tree could be included as a component of the visualization of hierarchical clusters that I propose in later chapters.

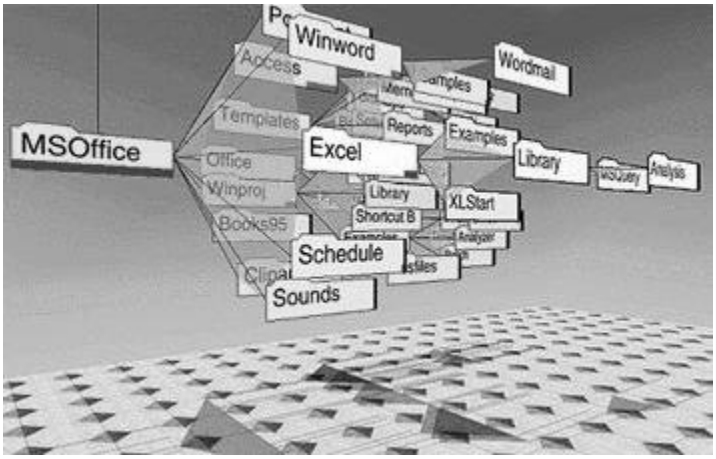


Figure 2-13: The cone tree for visualizing hierarchies, developed at Xerox PARC.

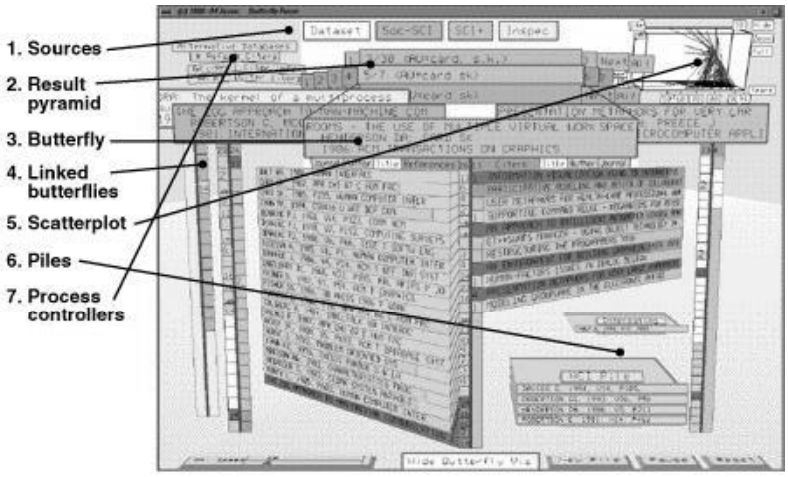


Figure 2-14: Butterfly user interface for searching citation links, developed at Xerox PARC.

Mackinlay *et al* have developed the innovative Butterfly visualization for searching citation links [Mack95], shown in Figure 2-14. A current document appears in the center of focus as the butterfly "body." Documents that cite and are cited by the current document for a given are visualized as separate butterfly "wings." The system also includes the automatic creation of citation graphs from database records,

asynchronous query processes to help alleviate user waiting, and embedded process control to provide fine control of focus.

While the Butterfly system visualizes citations for a single document only, my approach visualizes any number of documents simultaneously. Thus Butterfly is intended for small-scale interaction with document collections. In fact, a Butterfly-type approach could be added to the visualizations I propose in later chapters, as a method of navigation.

This chapter has given background and motivation for the approach I take to visual-based information retrieval, and has described related work. In the next chapter, introduce the idea of higher-order citation analysis, and its application to the visualization of document clusters.

Chapter 3

Clustering with Higher-Order Co-Citations

The importance of clustering in information retrieval is well known [Baez99]. Moreover, complex clustering results are often more easily understood through visualizations. While innovative information retrieval systems often employ link analysis for hypertext, the analysis output is usually presented as a simple linear list, with the only relationship among documents being query relevance. Useful query-independent analyses like clustering are usually not done, and/or the analyses are not enhanced with visualizations.

This chapter proposes a new approach to information retrieval based on visualizing clusterings of the query results. It is based on a generalization of the co-citation relationship between documents, in which the usual association between a pair of documents is generalized to an association among larger sets of documents. Because this generalized co-citation on involves larger sets of documents, it is seen as a higher-order co-citation. It is equivalent to association mining itemsets in the field of data mining.

This chapter proposes a methodology for computing inter-document distances that retains the simple pairwise structure while including higher-order co-citation information. As such, the new distances are a hybrid between pairwise and higher order. The hybrid distances offer greatly reduced complexity for the user, and allow the direct application of standard clustering algorithms. But the clusters resulting from these new distances are

much more consistent with frequently occurring itemset in comparison to standard distances.

I propose a new metric for comparing the clusterings resulting from various distance formulas, based on how well they match frequent itemsets. I apply the metric in a number of experiments using data sets extracted from the SCI (Science Citation Index). These experiments are designed to test the effects of the new hybrid pairwise/higher-order distances on clustering.

The first section of this chapter introduces the computation of distances from the co-citation relationship. Section 3.2 then describes hierarchical clustering, and the dendrogram visualization of hierarchical clusters. It also proposes a new augmentation of the dendrogram with glyphs for frequent itemsets. Section 3.3 describes a new itemset-matching metric for clusterings. Guided by the augmented dendrogram and itemset-matching clustering metric, Section 3.4 then introduces new hybrid pairwise/higher-order document distances for hierarchical clustering visualization of hyperlinked documents. To conclude the chapter, Section 3.5 experimentally tests the effects of hybrid distances on clustering for SCI data sets.

3.1 Co-Citation Distances

Hyperlink systems, e.g. the World Wide Web or science citations, can in general be modeled as directed graphs. A graph edge from one document to another indicates a link from first to second. However, it is also convenient to apply a matrix formulation for the development of link-based document distances. In fact, for actual implementation

this leads to the direct application of matrix data structures and operations usually found in programming languages.

In the matrix formulation, a binary *adjacency matrix* is formed which corresponds to the linkage graph. I take the convention that adjacency matrix rows are for citing documents and columns are for cited documents. Thus for adjacency matrix **A**, element $a_{i,j} = 1$ indicates that document i cites document j, and $a_{i,j} = 0$ is the lack of citation.

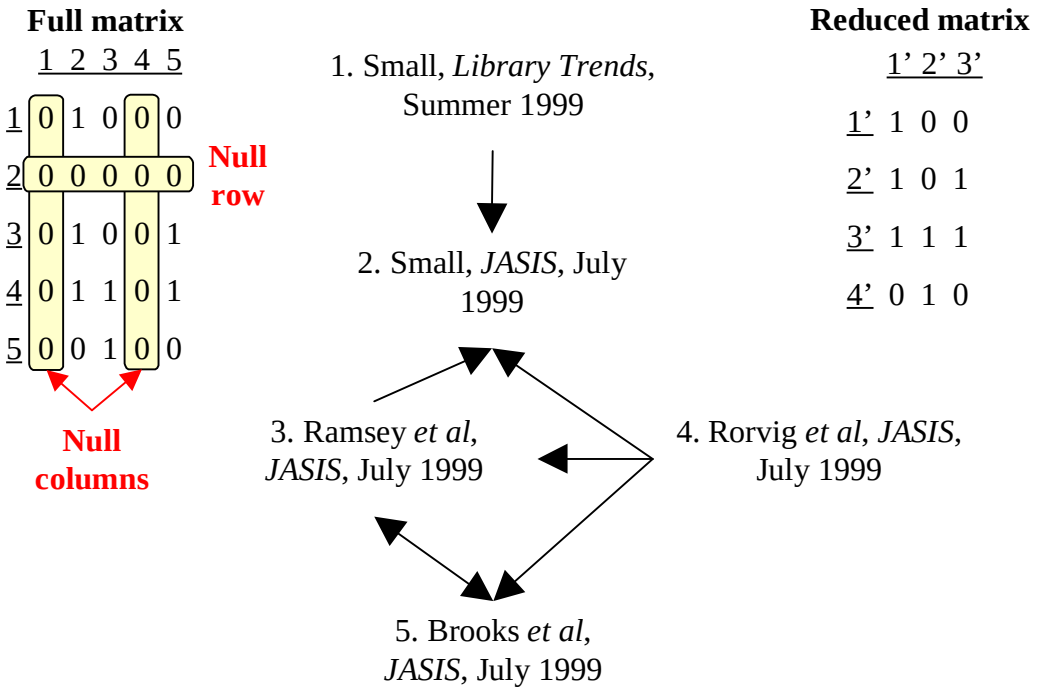


Figure 3-1: Full and reduced citation adjacency matrix from citation or hyperlink graph.

We may start from the full $n \times n$ adjacency matrix for n documents in a collection. However, some documents may cite no others in the collection, or correspondingly, some may be cited by no others in the collection. This corresponds to null (all zero) rows or columns, respectively, of the adjacency matrix. It is convenient to

reduce the full adjacency matrix by the removal of any null rows or columns, as shown in Figure 3-1. The resulting reduced adjacency matrix is in general no longer square.

If the citation graph is acyclic, e.g. the graph is partially ordered by time for published papers, it can be topologically sorted [Corm96]. The resulting adjacency matrix would then be triangular. Although not done here, this triangular structure could potentially be exploited in reducing the complexity of computing link-based distances.

A co-citation between 2 documents is the citing (or hypertext linking) of the 2 documents by another one, as shown in Figure 3-2 [Smal73]. A measure of similarity between a pair of documents can be defined as the number of documents that co-cite the pair. This is known as citation count. Taken over all pairs of documents, the co-citation count similarity serves as a compact representation of citation graph structure.

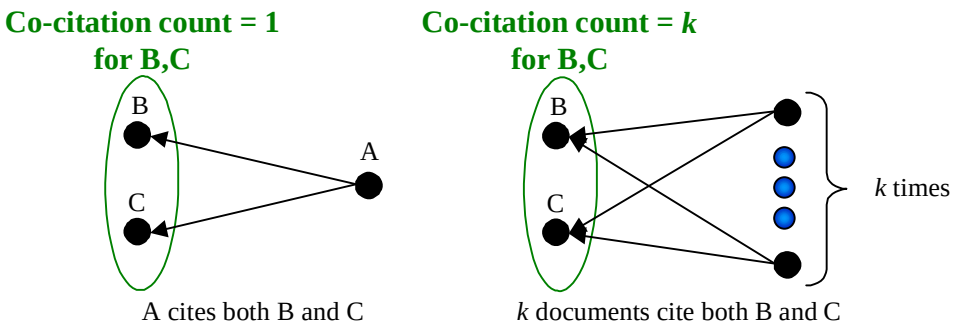


Figure 3-2: Co-Citation count.

In terms of the (reduced) adjacency matrix $\mathbf{A}$, 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

$$c_{j,k} = \sum_i a_{i,j} a_{i,k} = \mathbf{a}_j \bullet \mathbf{a}_k = \mathbf{A}^{\mathrm{T}} \mathbf{A} . \qquad (3.1)$$

Here $\mathbf{a}_j$ and $\mathbf{a}_k$ are column vectors of $\mathbf{A}$, i indexes rows, $\mathbf{A}^{\mathrm{T}}$ is the transpose of $\mathbf{A}$, and $\mathbf{x} \bullet \mathbf{y}$ is the vector dot (inner) product. Note that $a_{i,j}a_{i,k}$ 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

$$\hat{c}_{j,k} = \frac{c_{j,k} - \min(c_{j,k})}{\max(c_{j,k}) - \min(c_{j,k})}, \tag{3.2}$$

yielding the normalized count $\hat{c}_{j,k} \in [0,1]$. 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. When interpreted in terms of geometry, these similarities are called distances.

One way to convert similarities to dissimilarities (distances) is through the nonlinear operation of multiplicative inversion. An inversion formula that avoids singularities (division by zero) for normalized co-citation count $\hat{c}_{j,k} \in [0,1]$ is

$$d_{j,k} = \frac{1}{1 + \hat{c}_{j,k}}, \tag{3.3}$$

resulting in distance $d_{j,k}$ between documents j and k, normalized to $d_{j,k} \in [1/2,1]$. Another way to convert normalized co-citation count $\hat{c}_{j,k}$ to distance $d_{j,k}$ is the linear transformation

$$d_{j,k} = 1 - \hat{c}_{j,k}. \tag{3.4}$$

In this case, $d_{j,k}$ is normalized to $d_{j,k} \in [0,1]$.

Another possible measure of co-citation similarity between 2 documents is correlation. The most commonly applied is Pearson's product-moment correlation coefficient, or just Pearson's correlation. Pearson's correlation r for general variables x and y is defined as

$$\begin{aligned} r &= \frac{1}{n-1}\sum\left(\frac{x-\mu_x}{\sigma_x}\right)\left(\frac{y-\mu_y}{\sigma_y}\right) \\ &= \frac{\sum xy - \frac{1}{n}\left(\sum x\right)\left(\sum y\right)}{(n-1)\sigma_x\sigma_y}. \end{aligned} \tag{3.5}$$

Here μ_x and μ_y are means, σ_x and σ_y are standard deviations, and n is the number of observations.

Applying Eq. (3.5) to the citation adjacency matrix $\mathbf{A}$, $x \equiv a_{i,j}$ and $y \equiv a_{i,k}$, so the correlation $r_{j,k}$ for columns (cited documents) j and k is

$$\begin{aligned} r_{j,k} &= \frac{\sum_i a_{i,j}a_{i,k} - \frac{1}{n}\left(\sum_i a_{i,j}\right)\left(\sum_i a_{i,k}\right)}{(n-1)\sigma_j\sigma_k} \\ &= \frac{c_{j,k} - \frac{1}{n}\left(\sum_i a_{i,j}\right)\left(\sum_i a_{i,k}\right)}{(n-1)\sigma_j\sigma_k}. \end{aligned} \tag{3.6}$$

Here $c_{j,k}$ is the co-citation count, and σ_j and σ_k are standard deviations for columns j and k. We see that Pearson's correlation for 2 columns of the adjacency matrix is simply the co-citation count with zero-mean unit-variance columns. Correlation is bipolar, i.e. $r_{j,k} \in [-1,1]$. Because we seek a measure of similarity, we want correlation absolute value. Correlation absolute value is then converted to a distance via either

$$d_{j,k} = \frac{1}{1+\left|r_{j,k}\right|}, \tag{3.7}$$

or

$$d_{j,k} = 1 - \left| r_{j,k} \right| . \tag{3.8}$$

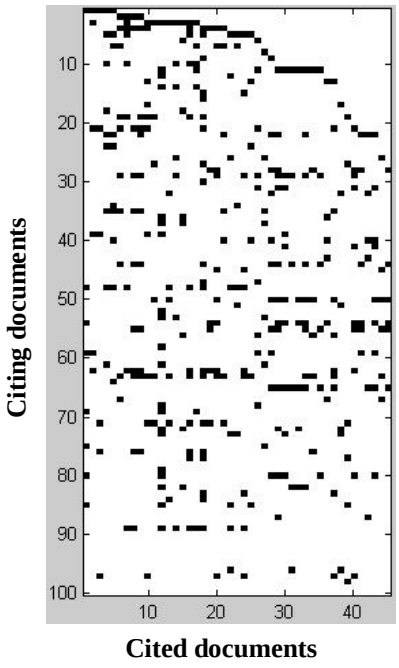
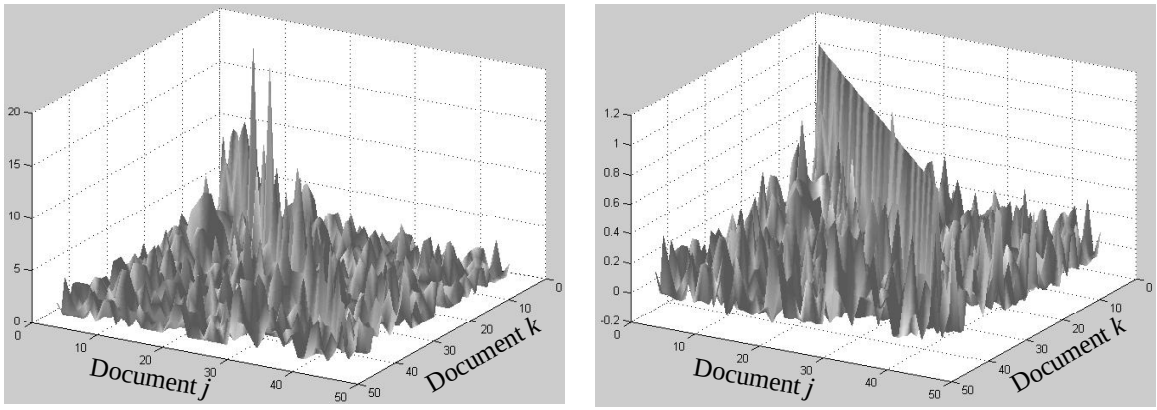


Figure 3-3: Citation matrix for SCI (Science Citation Index) "Microtubules" data set. Black indicates presence of citation, and white indicates lack of citation.

Let us examine an example computation of these co-citation-based distances. I begin with a query to the Science Citation Index, keyword "microtubules" for year 1999. I selected the first 100 documents returned from the query. These 100 documents cite a total of 3070 unique documents. I then retain only those cited documents that have been cited 6 or more times (this is a typical filtering in citation analysis, for retaining only the more frequently cited documents). Figure 3-3 shows the resulting 100×45 citation matrix.

Figure 3-4 shows co-citation counts and cited document correlations for the "Microtubules" data set. These are computed with Eq. (3.1) and Eq. (3.6), respectively. The main diagonal of the correlations is unity, while the main diagonal of the co-citation counts is the citation count for each cited document. The correlation values are strongly biased toward positive values, although there are some negative values. Figure 3-5 shows the minimum values for each column of the correlation matrix.



Co-citation count

Cited document correlation

Figure 3-4: Co-citation counts and cited document correlation for SCI "Microtubules" data set.

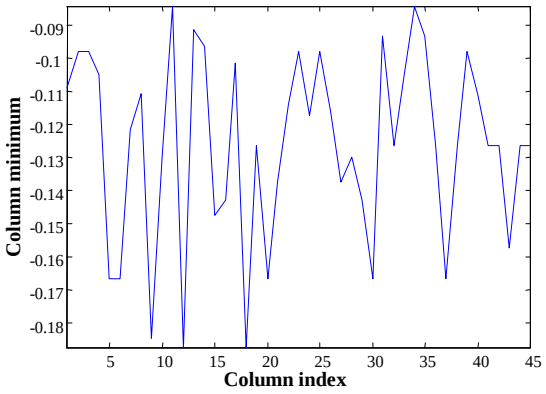


Figure 3-5: Correlation matrix column minimum values for SCI "Microtubules" data set.

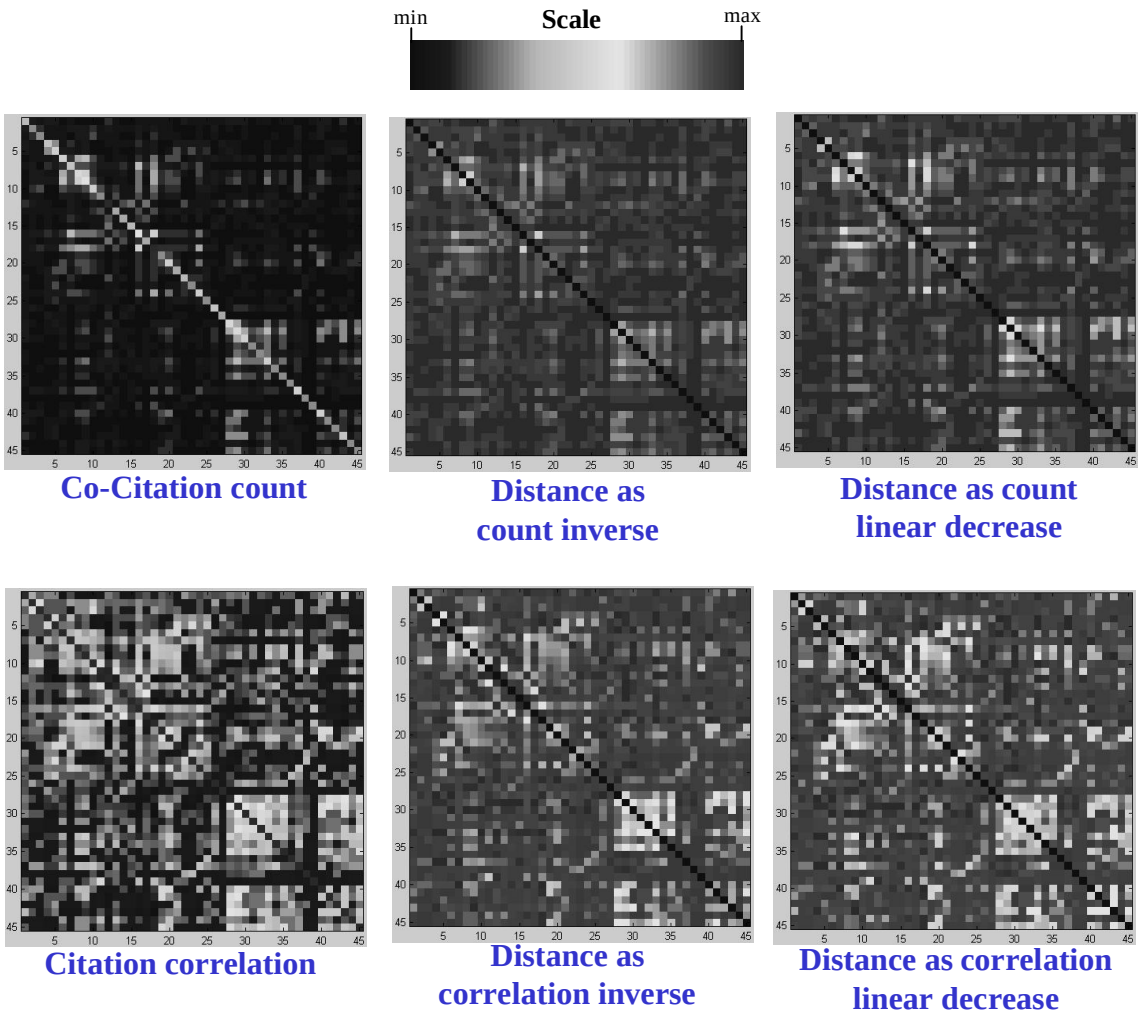


Figure 3-6: Co-citation counts, correlations, and corresponding multiplicative and additive inverse distances for SCI "Microtubules" data set.

Figure 3-6 shows 4 different citation-based distance matrices for the 45 cited documents in the "Microtubules" data set. The distances are computed from both co-citation counts and cited document correlations, also shown in the figure. Distances are computed from co-citation counts or correlations as either multiplicative or additive inverse, via Eq. (3.3) and Eq. (3.4) for co-citation counts, or Eq. (3.7) and Eq. (3.8) for correlations.

From Figure 3-6, we see that the general structure of the 4 distance matrices is the same. In particular, larger as well as smaller distances generally appear in the same place within each matrix. For both co-citations and correlations, the additive inverse similarity-to-distance conversion yields larger distance variances than the multiplicative inverse conversion.

I now test each of the 4 distance matrices in Figure 3-6 for being *metric*. In particular, I test whether they obey the triangle inequality $d(a,c) \leq d(a,b) + d(b,c)$ that holds for metric distances. For each matrix, I test the inequality for every possible triangle (set of 3 matrix elements). Metricity is important because deviations from the triangle inequality cannot be drawn for visualization.

The inequality holds for every triangle for each of the 3 distance matrices "count multiplicative," "count additive," and "correlation multiplicative." For the "correlation addive" matrix, the inequality holds for all but 5 triangles (out of 42570 total). I detect no particular pattern for the ones that did not hold.

Correlations have been previously proposed [Whit98] as measures of similarity between cited documents. But there are some conceptual as well as practical problems with these correlations, in contrast to co-citation counts. The argument given in favor of correlations is to "preserve patterns of co-citation over all [citing] documents." In other words, correlation measures the similarity of cite/no-cite bit patterns for a cited document (citation matrix column) pair.

A conceptual problem with correlation is that "cite" bits (ones) of the citation matrix are treated the same as "no-cite" bits (zeros). In particular, if 2 columns (cited documents) are exactly the same, the correlation is unity, regardless of the actual number

of one bits versus zero bits. For example, 2 columns with only a single one bit (in the same row) yield the same maximum correlation as 2 columns comprised of all one bits. This disregards the actual number of co-citations. In short, correlations remove column means and variances, which in this case are important information.

An important point is that in this case ones in the citation matrix are not the same type of information as zeros. Ones give an association between citing and cited papers. Zeros are not statements about a lack of association – they are more like a lack of information about an association. So the presence of a one is much more important than the presence of a zero in terms of citation-based document similarity. Co-citation counts measure only the simultaneous presence of ones.

More practically, correlations have higher computational complexity than co-citation counts, since they must remove means and normalize variances. Also, I show in Chapter 4 that vertices of minimum spanning trees resulting from correlations are more difficult to position for visualization. For pairs of documents with the same co-citation count but with different means and/or variances, the removal of the means and normalization of the variances results in different similarities for different pairs. This finer similarity granularity leads to trees with generally lower-degree vertices. The result is larger numbers of local minima with respect to the vertex-positioning algorithm.

3.2 Hierarchical Clustering and Dendrograms

Clustering plays a central role in information retrieval. In classical work, clustering based on co-citation similarity is known to correspond well to individual fields of knowledge [Garf79]. For information retrieval, results from simple keyword queries

can be clustered into more refined topics. Co-citation-based clustering provides a narrowing of search results, by allowing the user to focus on documents in pertinent clusters only. This helps alleviate the potentially tedious task of manually reviewing large lists of search results. Also, co-citation analysis can broaden search results by providing alternative documents linked by co-citation.

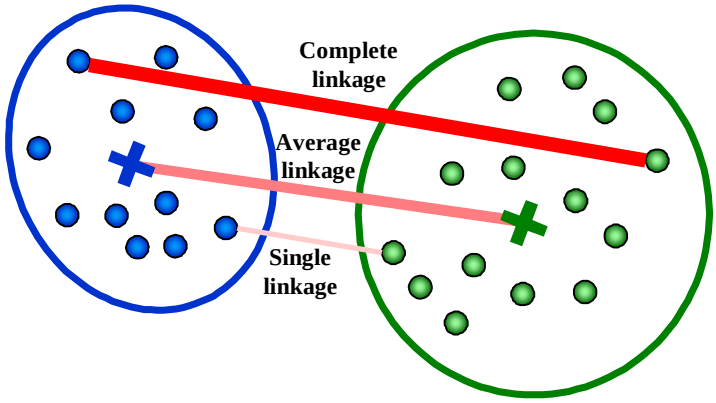


Figure 3-7: Inter-cluster distances for single-linkage, average-linkage, and complete-linkage.

Three important heuristics for clustering are *single-linkage*, *average-linkage*, and *complete-linkage*. These heuristics are agglomerative, at each step merging clusters that have the closest distance between them. Arguments have been given for all 3 heuristics in terms of desirable clustering characteristics [Vena94].

Single-linkage, average-linkage, and complete-linkage clustering are illustrated in Figure 3-7. For single-linkage, the measure of distance between 2 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-linkage, and complete-linkage correspond to weak, intermediate, and strong clustering criteria, respectively.

The *dendrogram* is a tree visualization of a clustering. It reflects hierarchical clustering. That is, each cluster is comprised of sub-clusters, iterated down to the level of individual documents (clusters of one object). Leaves of the dendrogram tree are individual documents, at the lowest level of the hierarchy. Non-leaf nodes represent the merging of 2 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 Figures 3-8 and 3-9.

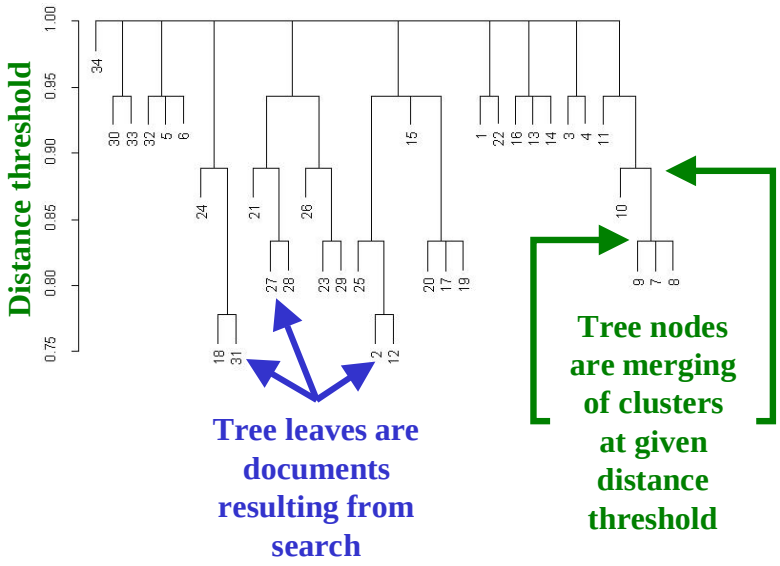


Figure 3-8: Dendrogram tree for visualizing clustering hierarchy.

For a given hierarchical clustering, the clusters resulting from a given threshold distance can be readily determined from the dendrogram. 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.

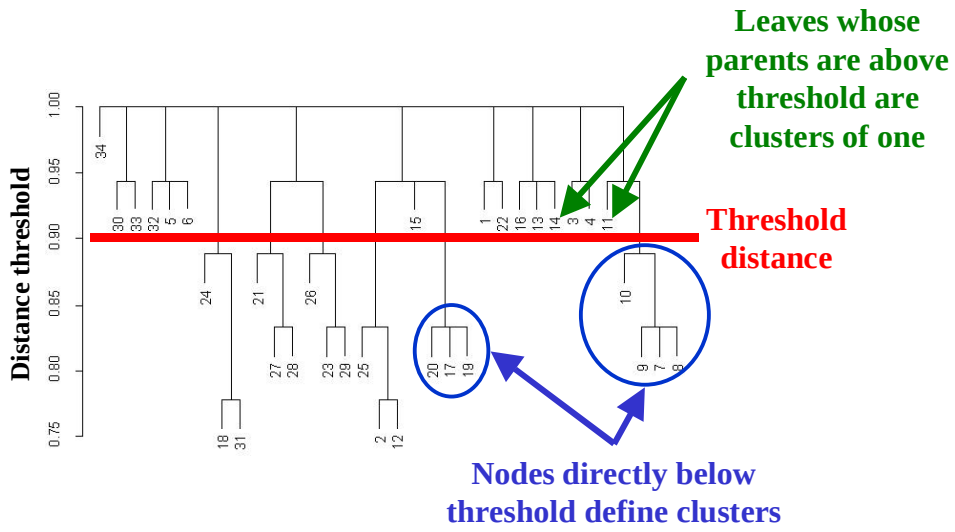
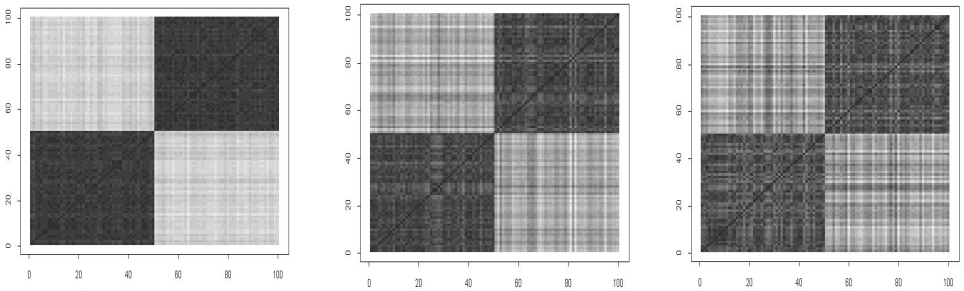


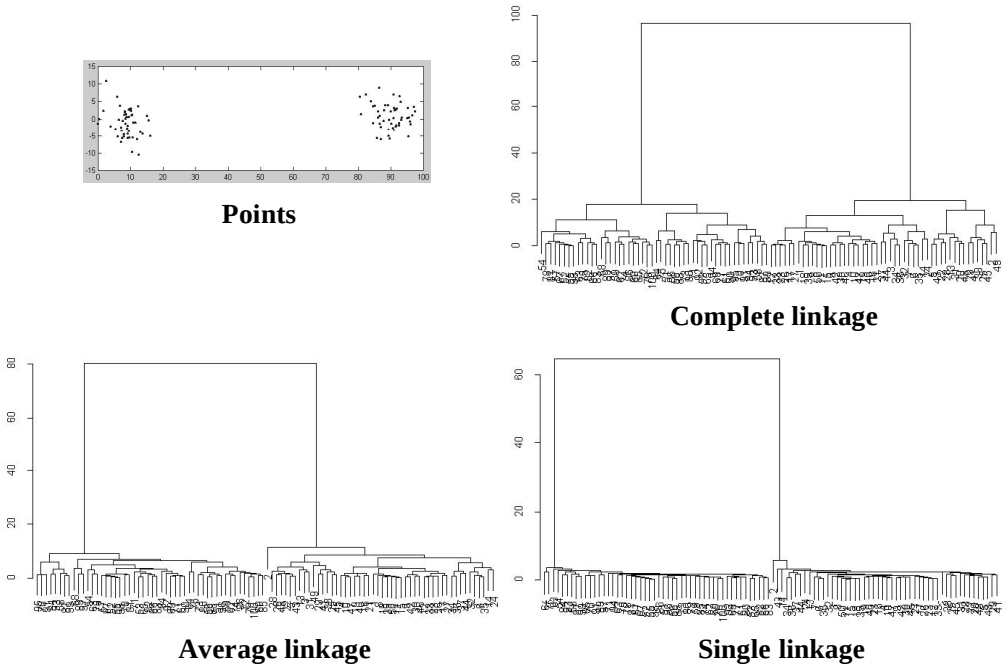
Figure 3-9: Interpreting clusters from dendrogram for given clustering threshold distance.

Figures 3-11 through 3-13 compare the 3 clustering heuristics (complete-linkage, average-linkage, and single-linkage) as inherent clusters in the data become less well separated. Here I apply synthetic data, for which the inherent clusters and their separation are known. In particular, the points are samples from 2-dimensional Gaussian distributions, with distribution means defining cluster separations. Figure 3-10 shows the corresponding distance matrices for the points.



∆ mean = 80 **∆mean = 30** **∆mean = 20**

Figure 3-10: Distance matrices for Figures 3-11 through 3-13, in which mean of Gaussian distribution defines cluster separations.



Points **Complete linkage**

Average linkage **Single linkage**

Figure 3-11: Dendrograms for 2 very well separated clusters (∆mean = 80).

For well-separated clusters, the 3 heuristics perform similarly. The inherent 2 clusters result for a wide range of threshold. While the largest absolute threshold range occurs for complete-linkage (about 80 units versus about 70 and 60, respectively), the

largest relative threshold range occurs for single-linkage. In particular, single-linkage gives a sharper transition from a complete collection of single -point clusters to 2 clusters.

But the situation changes once the clusters are close enough together. Single-linkage causes the 2 clusters to coalesce to a single cluster, through points that are near points in the other cluster. This phenomenon is known as “chaining.” Threshold values small enough to produce multiple clusters cause distant points to form single-point clusters before 2 main clusters are obtained.

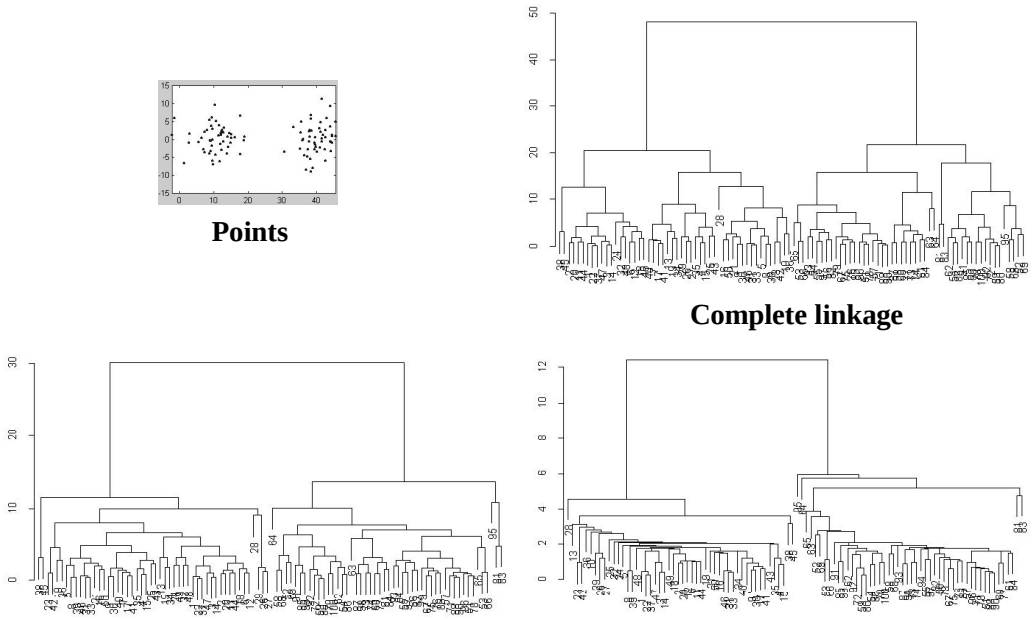


Figure 3-12: Dendrograms for 2 well separated clusters (Δmean = 30).

Figure 3-14 shows single-linkage chaining more explicitly. The synthetic data points form 2 main Gaussian clusters, with a linear chain of points between them. In addition to the points, the figure shows the corresponding distance matrix, and dendrograms for complete-linkage, average-linkage, and single-linkage clustering.

Complete-linkage and average-linkage are able to separate the 2 clusters, despite the chain. For significant ranges of clustering threshold, 2 clusters result, with the chain points being divided between the 2 clusters. For single-linkage, lowering the threshold only causes individual points to separate from the main cluster. Two somewhat large clusters form only after the threshold is reduced below the distance between the chain points. But this occurs only after many of the more distant points have separated into single-point clusters.

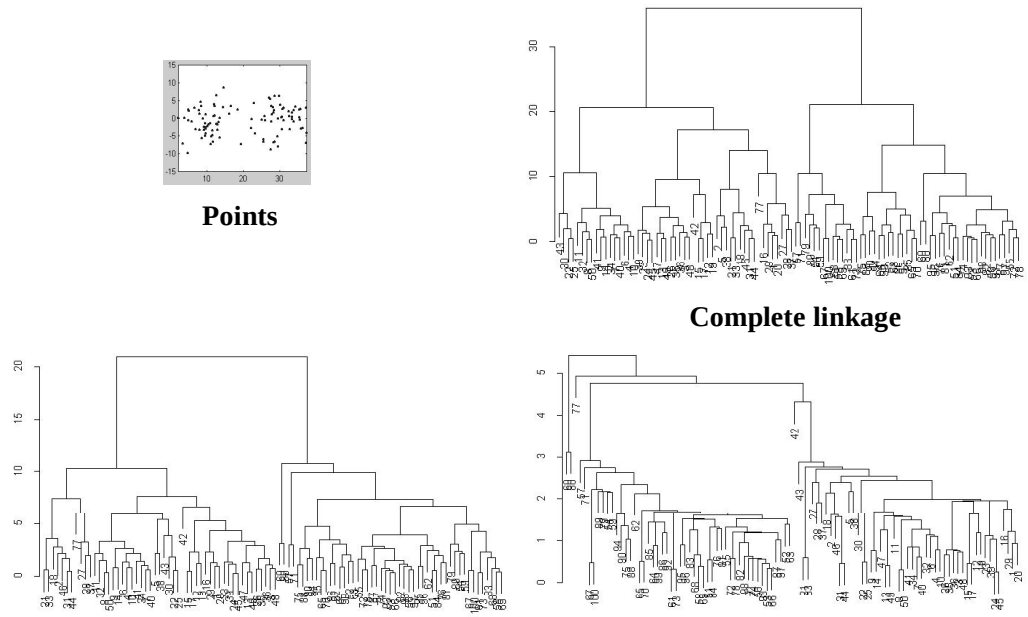


Figure 3-13: Dendrograms for 2 poorly separated clusters (Δmean = 20).

Figures 3-15 through 3-18 show hierarchical clustering results for the 45 documents cited 6 or more times within the SCI "Microtubules" data set. The figures correspond to distances defined by Eqs. (3.3), (3.4), (3.7), and (3.8), respectively.

Clustering results for the 2 distances based on co-citation count are nearly the same. The only differences are minor ones for average-linkage clustering.

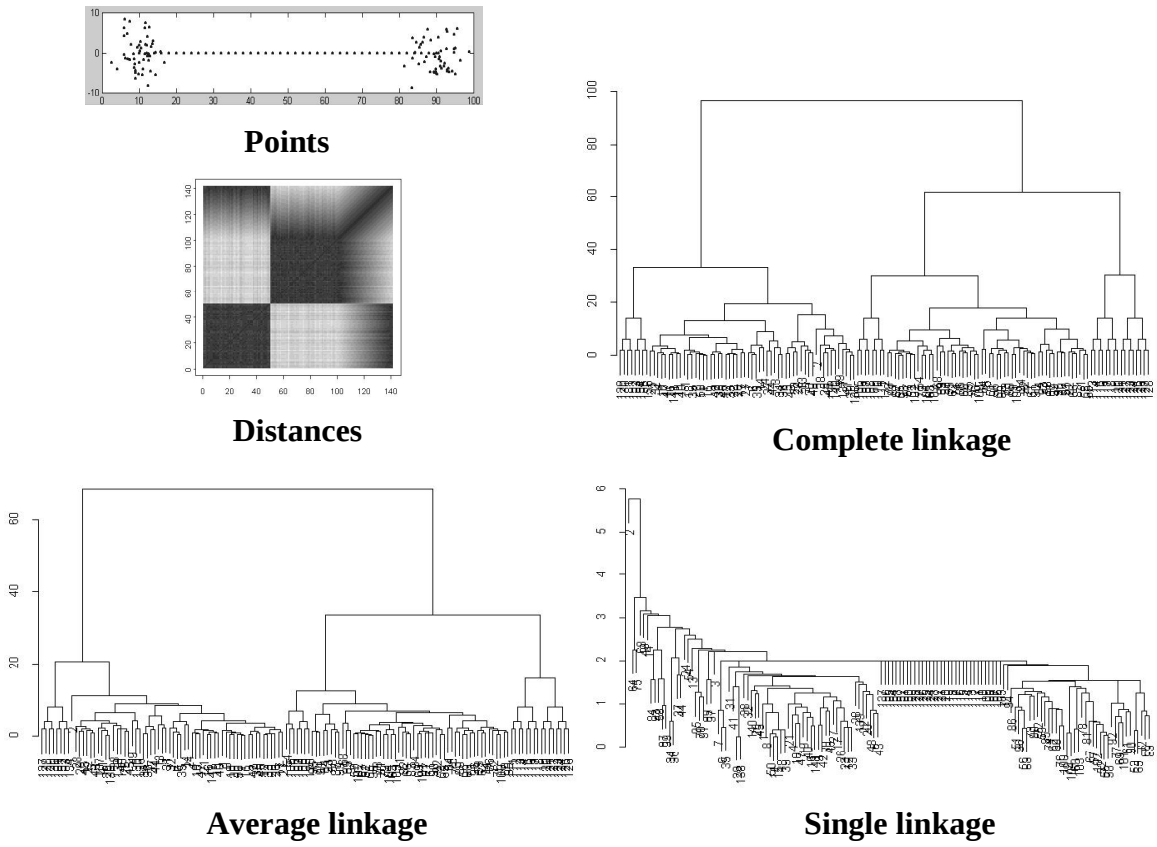
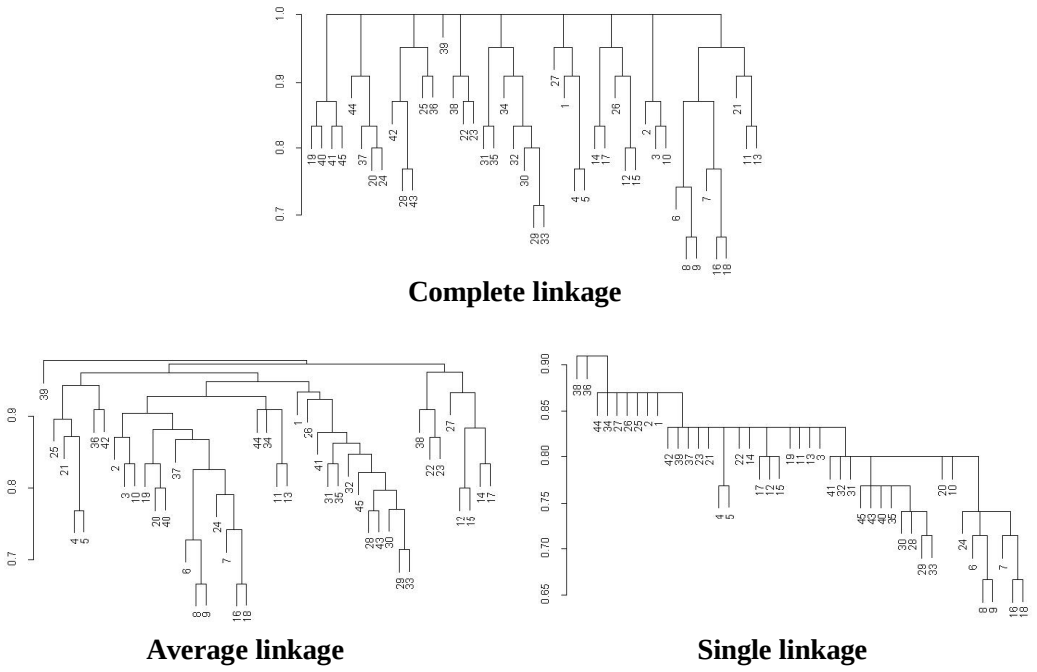


Figure 3-14: Dendrograms for 2 clusters with chain of points between them.

While the general clustering structure is the same between count-based and correlation-based distances, many of the details differ. A more global difference is that for count-based distances, multiple tree nodes occur at the same threshold value, corresponding to multiple document pairs with the same distance. In contrast, for correlation-based distances tree nodes are generally all at different threshold values. This is a consequence of the mean removal and variance normalization for correlations.

Generally, there are more differences between the 2 correlation-based distances than between the 2 count-based ones. For single-linkage, clustering results for the 2 correlation-based distances are the same. For complete-linkage and average-linkage, the lower-level clusters are generally the same, but there are significant upper-level differences.

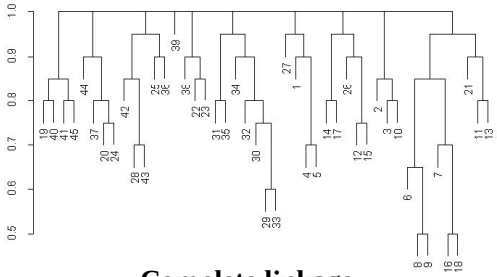


Complete linkage

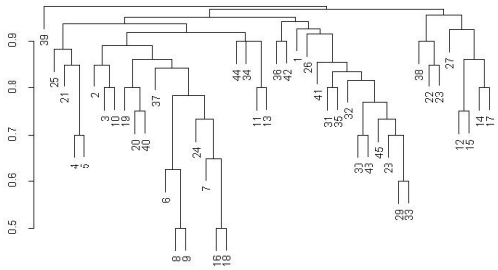
Average linkage **Single linkage**

Figure 3-15: Dendrograms for "count inverse" distances for "Microtubules" data set.

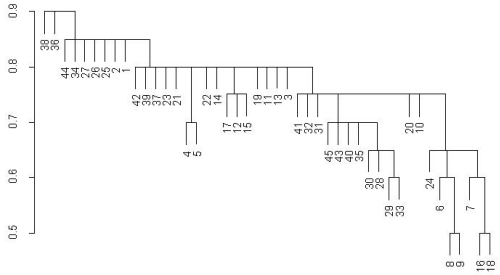
For a given distance method, results from the 3 types of clustering certainly differ. This suggests that these documents are not inherently distributed as well-separated clusters. In particular, there is evidence of single-linkage "chaining." This is in direct contrast to the generally held notion that typical document collections have well defined clusters, so that single-linkage is adequate [Garf79]. However, at least one author has suggested that single-linkage clustering may be inadequate for citation analysis [Smal93].



Complete linkage

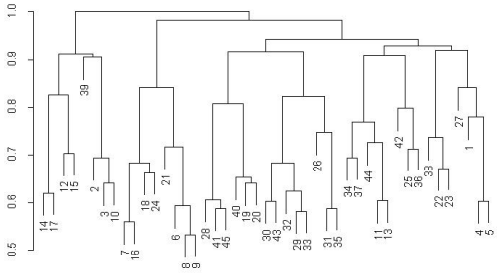


Average linkage

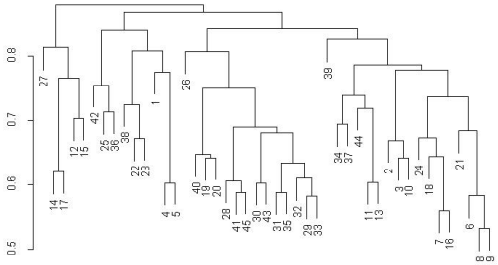


Single linkage

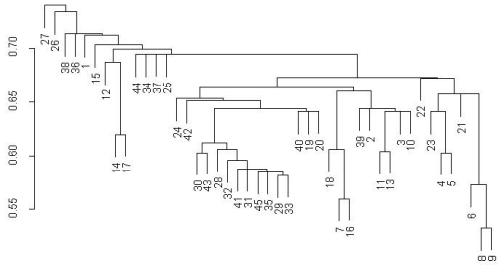
Figure 3-16: Dendrograms for "count additive" distances for "Microtubules" data set.



Complete linkage

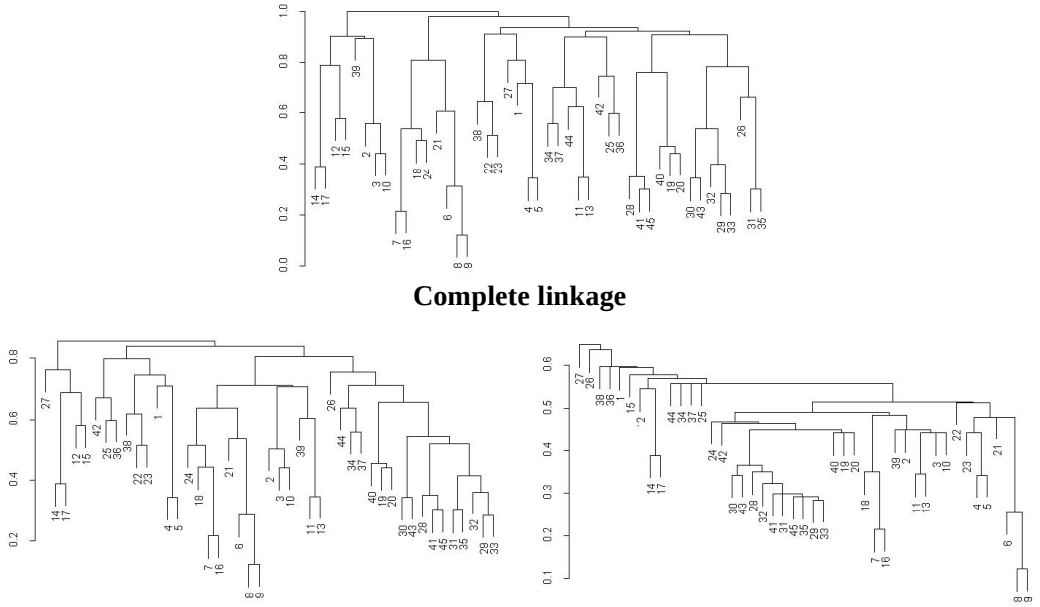


Average linkage



Single linkage

Figure 3-17: Dendrograms for "correlation inverse" distances for "Microtubules" data set.



Complete linkage

Average linkage **Single linkage**

Figure 3-18: Dendrograms for "correlation additive" distances for "Microtubules" data set.

To obtain a rough characterization of a statistical model for co-citation-based distances, I compare clusterings for 2 different distributions. Each distribution is 2-dimensional for ease of visualization, but one can imagine that the distributions are projections onto 2 dimensions of the higher dimensional graph spaces characteristic of co-citations. Samples of the distributions, along with corresponding distances and dendrograms are shown in Figures 3-19 and 3-20.

The first distribution is uniformly distributed in the plane. The 2nd distribution is a Levy flight [Mand83], a random walk model in which each step has uniformly distributed direction, with magnitude distributed as a decaying exponential. The Levy flight is fractal, having no characteristic scale. It is inherently comprised of hierarchical clusters over all scales.

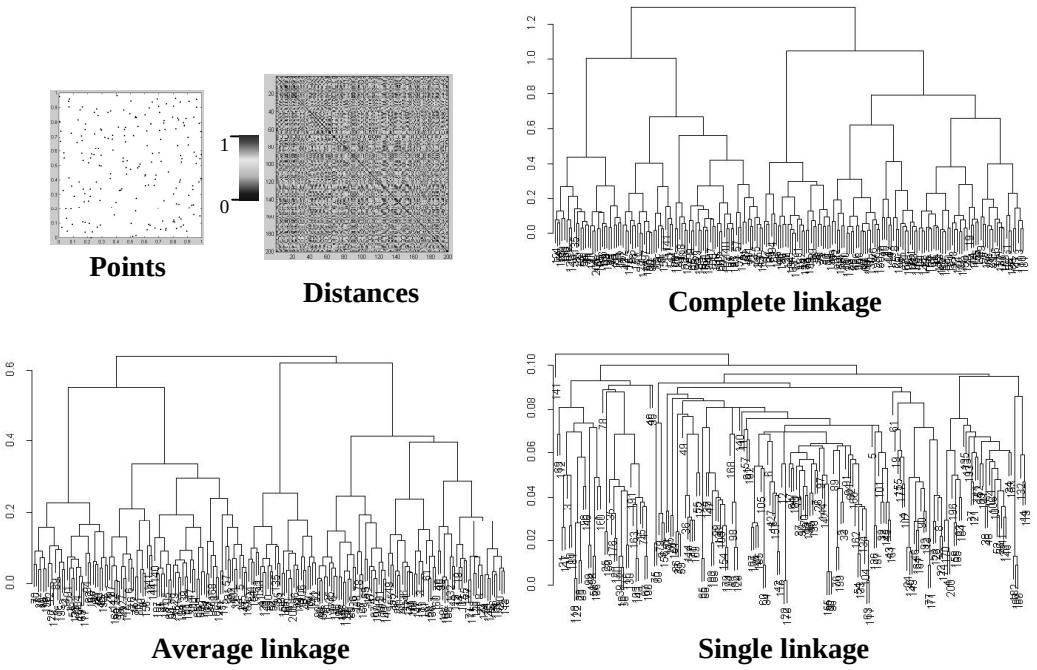


Figure 3-19: Hierarchical clustering for uniform random points.

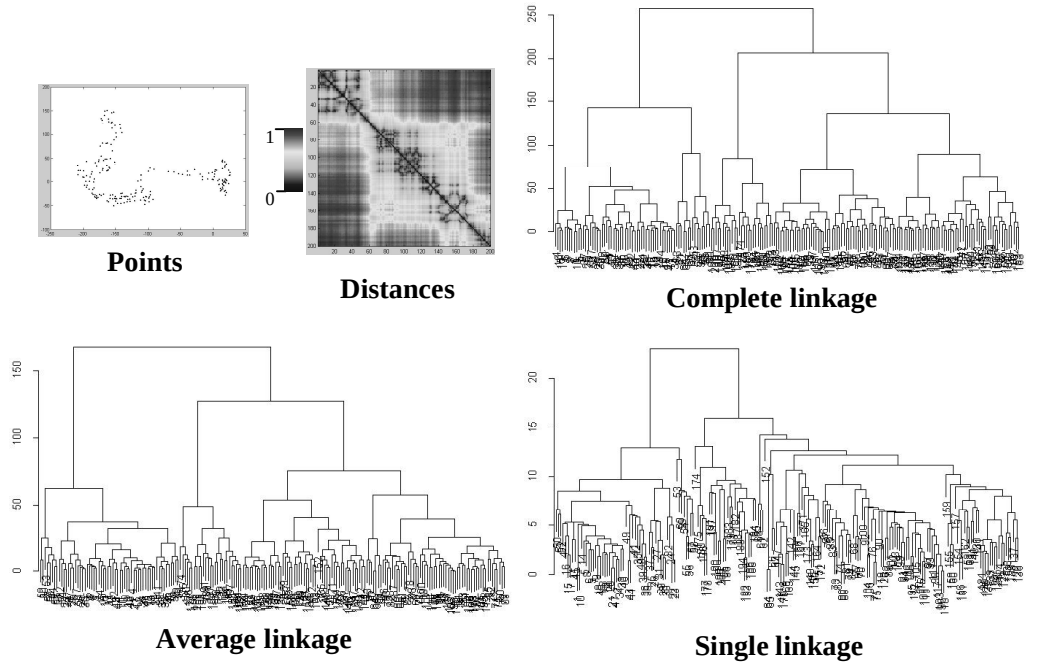


Figure 3-20: Hierarchical clustering for fractal random points.

Comparing the models in Figures 3-19 and 3-20 to the real citation data in Figures 3-6 and Figures 3-15 through Figures 3-18, the distance and clustering structures for the citation data correspond more closely with the fractal model versus the uniform distribution. In fact, a fractal distribution for co-citation-based cluster sizes has been previously reported [VanR90]. In Section 3.4, I provide further evidence for the fractal nature of co-citations, particularly as generalized to higher-order co-citations.

3.3 Itemset-Matching Clustering Metric

A central component in classical citation analysis is clustering based on co-citations as a measure of similarity. In the case of co-citations, an association is made between 2 documents according to the number of times they are co-referenced, i.e. through hypertext links or literature citations. The purpose of clustering is to form larger sets of documents that are more strongly associated with one another than they are to documents outside the cluster.

Traditional citation analysis typically applies single-linkage clustering, because of its lower computational complexity. But as we have seen, because of its very weak clustering criterion, single-linkage has problems unless the data are inherently well clustered. Given the improved performance of computing machines, it becomes feasible to apply stronger clustering criteria in citation analysis.

Figure 3-21 shows the weak clustering criterion of single-linkage, and how it relates to citations. For the 3 cited documents, there are $\left(n^2 - n\right)/2 = 3$ possible co-citation similarities. As the example shows, only 2 of these similarities need to exceed

the clustering threshold for the 3 documents to be considered a cluster, as long as they share a common document between the 2 pairs.

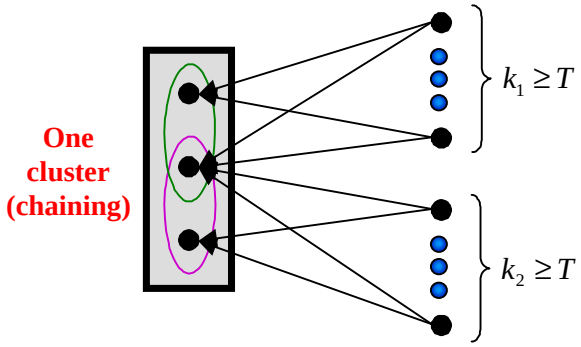


Figure 3-21: Single-linkage chaining for co-citation similarities.

In contrast, for the stronger clustering criterion of complete linkage *all* similarities for the 3 pairs need to exceed the threshold before the documents constitute a single cluster. This is shown in Figure 3-22. But notice that for this example, there is not even one document that cites all 3 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 the co-citation similarity in which sets of cardinality above 2 are considered for co-citation, as shown in Figure 3-23. That is, we define a similarity among a set of cited documents that is based on the number of times all the members of the set are simultaneously cited. Because the similarity is among more than 2 documents, we consider it to be higher order than pairwise.

We could then specify a threshold value for these higher-order similarities to identify sets whose similarities are sufficiently large. For our example, the only way the

3 cited documents could be considered a sufficiently similar set is if all 3 of them are cited more than the threshold number of times. These higher-order co-citations are known in the field of association mining as *itemsets* [Agra93]. Itemsets whose members are sufficiently similar (through what is called itemset *supports*) are known as *frequent* itemsets.

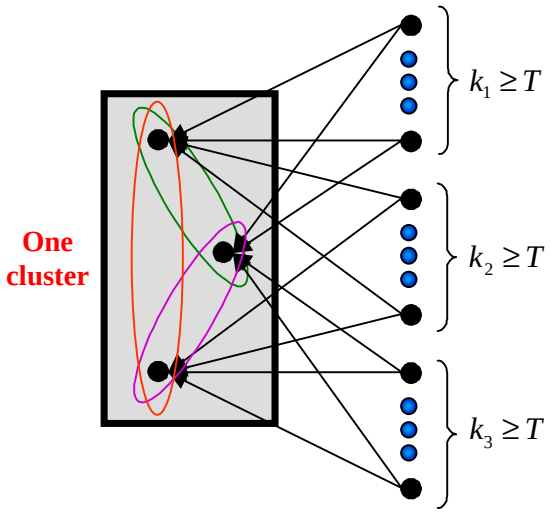


Figure 3-22: Stronger clustering criterion for complete-linkage with co-citation similarities.

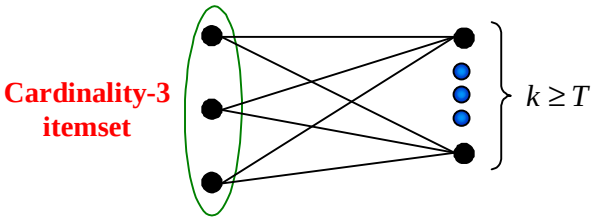


Figure 3-23: Higher-order co-citation (association mining itemset) is an even stronger association than complete-linkage cluster.

Another benefit of higher-order co-citations (association mining itemsets) 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.

Association mining is perhaps best known for its application to supermarket purchases, the so-called “market basket” problem. The associations are made among supermarket items based on how frequently they are purchased together. Association mining is a subfield of data mining, also known as knowledge discovery in databases.

An analogy can be made between supermarket purchases and document citations. The supermarket items are analogous to documents, and the purchase of items is analogous to the citation of documents. Thus association mining is applicable to citation analysis, forming associations among groups of documents that are frequently cited together. This mining would enhance understanding of the structure of document citations, making explicit those associations that may otherwise go unnoticed. Additional insight could be gained by forming citation associations at levels other than individual documents, for example authors, publications, institutions, or countries.

The analogy between supermarket purchases and document citations is not perfect. In the former, the entities making purchases (people) are distinct from the entities they purchase (items), whereas in the latter there is no such distinction (they are all documents). In the case of citations, there thus exists certain kind of symmetry in which clustering may be applied to either cit*ing* or cit*ed* documents. This is equivalent to the symmetry between co-citations and “bibliographic coupling” within traditional

citation analysis [Garf79]. This is handled mathematically by transposing the citation adjacency matrix.

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 $j_1, j_2, \ldots, j_{|I|}$, its scalar support $\zeta(I)$ is

$$\zeta(I) = \sum_i a_{i,j_1} a_{i,j_2} \cdots a_{i,j_{|I|}} = \sum_i \prod_{\alpha=1}^{|I|} a_{i,j_\alpha} , \tag{3.9}$$

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.

As before, higher-order co-citation similarities can be normalized as

$$\hat{\zeta}(I) = \frac{\zeta(I) - \min[\zeta(I)]}{\max[\zeta(I)] - \min[\zeta(I)]}, \tag{3.10}$$

so that $\hat{\zeta}(I) \in [0,1]$. They are then converted to dissimilarities via either multiplicative inversion

$$d_I = \frac{1}{1 + \hat{\zeta}(I)}, \tag{3.11}$$

normalized to $d_I \in [1/2, 1]$, or additive inversion

$$d_I = 1 - \hat{\zeta}(I), \tag{3.12}$$

normalized to $d_I \in [0,1]$.

By a fundamental property of sets, members of an itemset must be unique, that is

$$j_1 \neq j_2 \cdots \neq j_{|I|}. \tag{3.13}$$

Also, the extension from pairs of documents to sets of arbitrary cardinality means there is itemset overlap, that is, itemsets are non-disjoint. Such overlap 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 the set of items. The subsets form a partial ordering, under the ordering relation of set inclusion. This is illustrated in Figure 3-24, via the so-called Hasse diagram for visualizing partial orderings. The diagram shows the itemset lattice (excluding singletons and the empty set) for a set of 4 documents.

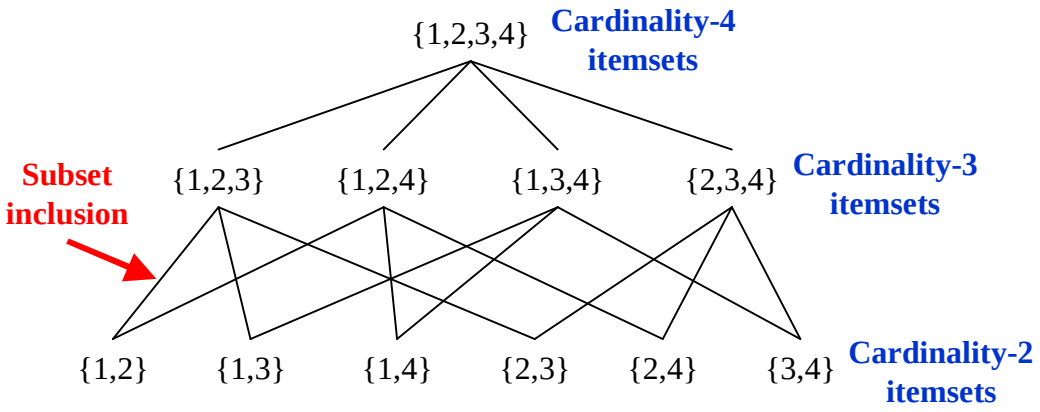


Figure 3-24: Itemset lattice for a set of 4 documents, visualized with Hasse diagram.

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

$$\binom{n}{|I|} = \frac{n!}{|I|!(n-|I|)!}, \tag{3.14}$$

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

Another data structure that has been recently proposed for itemsets is the itemset tree [Hafe99]. The itemset tree represents a compromise between the computational expense of traversing the database multiple times, and the storage expense for maintaining the complete itemset lattice. It addresses the need of association mining for dynamic data such as the World Wide Web.

I now compare frequent itemsets to graph-theoretic clustering for a data set from the SCI (Science Citation Index). I do the comparison by augmenting the dendrogram with members of frequent itemsets. This is the first time that such an augmentation has been proposed. The augmentation is feasible when the data set is sufficiently small, such that the combinatorial explosion of the number of itemsets described in Eq. (3.14) is manageable.

For the example, I do an SCI query with keyword “wavelet*” for the year 1999. The first 100 documents returned by the query cite 1755 documents. I filter these cited documents by citation count, retaining only those cited 3 or more times, resulting in a set of 34 highly cited documents.

I then compute complete-linkage, average-linkage, and single-linkage clusters and frequent itemsets for the set of highly cited documents. Here I apply Eqs. (3.1), (3.2), and (3.4) for computing co-citation based distances for clustering. The resulting augmented dendrogram is shown in Figure 3-25. The dendrogram is augmented by the addition of glyphs for members of frequent itemsets, added at the corresponding tree leaves.

The dendrogram in Figure 3-25 is augmented with frequent itemsets of cardinality 4. The single most frequent cardinality-4 itemset has a support of 3, that is, the set of 4

documents was cited simultaneously 3 times. The next most frequent cardinality-4 itemsets are 5 itemsets having a support of 2.

Overall, the itemsets are more consistent with complete linkage than with single linkage. This is not surprising, since complete linkage has a much greater clustering strength. Results for average linkage are similar to complete linkage, so I omit average linkage from the discussion. Table 3-1 gives brief bibliographic data for the set of highly cited documents being clustered.

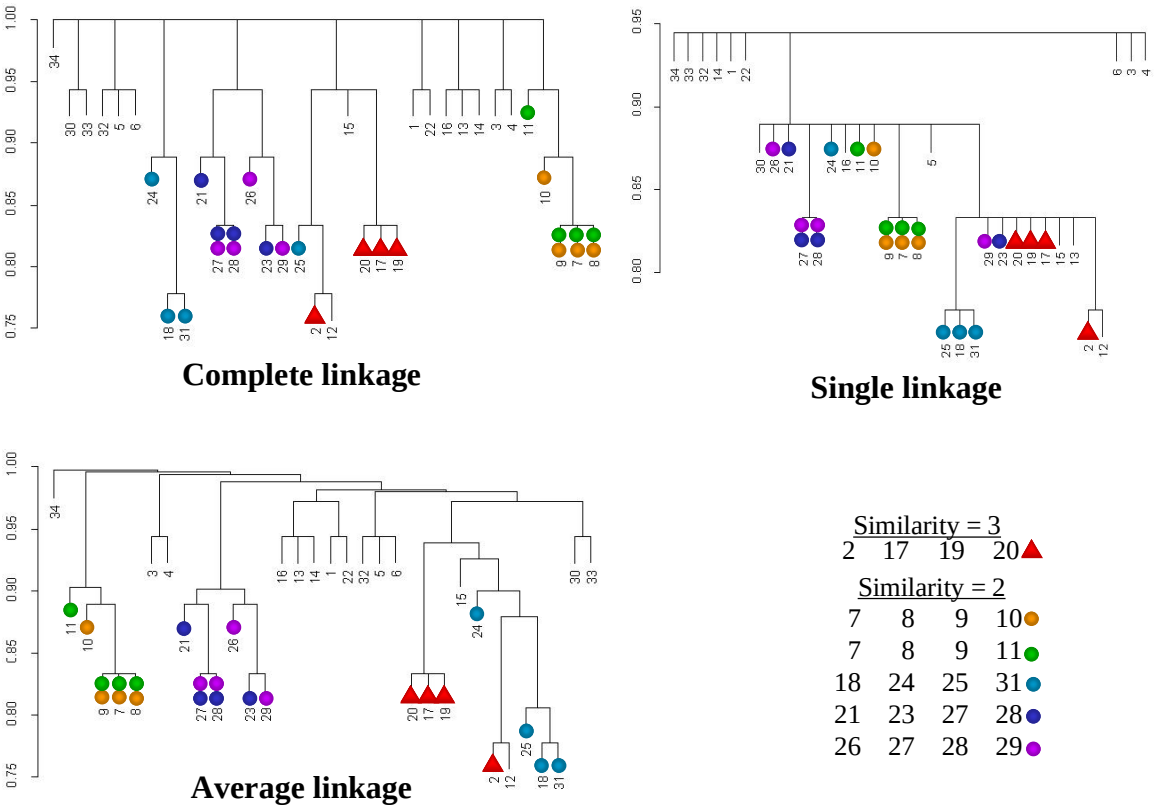


Figure 3-25: Clustering versus frequent itemsets for "Wavelets (1-100)" data set.

The cardinality-4 itemset of similarity 3 is {2, 17, 19, 20}▲. For complete linkage, documents 17, 19, and 20 of the itemset are a possible cluster. These are 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 paper by Mallat) in this cluster would require the inclusion of documents 12, 15, and 25, which are not in the itemset. These 3 additional documents are another foundational paper by Mallat and 2 foundational papers by Daubechies.

For single linkage, the situation is even worse. The itemset {2, 17, 19, 20}▲ is possible within a cluster only by including 8 other documents. I interpret this as being largely caused by single linkage chaining. The application of clustering to mere pairwise co-citation similarities is insufficient for ensuring that itemsets of larger cardinality appear as clusters, even with complete-linkage.

Because of the degree of overlap for the cardinality-4 itemsets {7, 8, 9, 10}● and {7, 8, 9, 11}●, they are likely the 5-itemset {7, 8, 9, 10, 11}. These 5 papers are largely foundational. In any event, the combined 2 itemsets are a complete-linkage cluster. But for single-linkage, 24 other documents would need to be included in order for the 2 itemsets to be a cluster. Again, pairwise clustering is a necessary but insufficient condition for frequent itemsets.

We have a similar situation for the cardinality-4 itemsets {21, 23, 27, 28}● and {26, 27, 28, 29}●, though with a lesser degree of itemset overlap. These papers are more technological, in particular they are applications of wavelets in image coding.

For the cardinality-4 itemset {18, 24, 25, 31}●, 3 of the papers are by Donoho, who works in wavelet-based statistical signal estimation for denoising. These 3 papers are a complete-linkage cluster, as well as a single-linkage cluster. The remaining document in the cardinality-4 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.

Table 3-1: Document indices and bibliographic details for "Wavelets 1999 (1-100)" data set.

	1	65	ANTONINI M, 1992, IEEE T IMAGE PROCESS, V1, P205
▲	2	75	MALLAT SG, 1989, IEEE T PATTERN ANAL, V11, P674
	3	118	MALLAT S, 1992, IEEE T PATTERN ANAL, V14, P710
	4	119	MALLAT SG, 1989, IEEE T ACOUST SPEECH, V37, P2091
	5	156	CHUI CK, 1992, INTRO WAVELETS
	6	158	FARGE M, 1992, ANNU REV FLUID MECH, V24, P395
● ●	7	165	AKANSU AN, 1992, MULTIRESOLUTION SIGN
● ●	8	166	CHEN CF, 1997, IEE P-CONTR THEOR AP, V144, P87
● ●	9	170	HSIAO CH, 1997, MATH COMPUT SIMULAT, V44, P457
●	10	173	STRANG G, 1989, SIAM REV, V31, P614
●	11	180	DAUBECHIES I, 1990, IEEE T INFORM THEORY, V36, P961
	12	185	DAUBECHIES I, 1988, COMMUN PUR APPL MATH, V41, P909
	13	187	MALLAT S, 1992, IEEE T INFORM THEORY, V38, P617
	14	243	STRANG G, 1996, WAVELETS FILTER BANK
	15	268	MALLAT SG, 1989, T AM MATH SOC, V315, P69
	16	273	VETTERLI M, 1995, WAVELETS SUBBAND COD
▲	17	313	BARCLAY VJ, 1997, ANAL CHEM, V69, P78
●	18	317	DONOHO DL, 1994, BIOMETRIKA, V81, P425
▲	19	320	MITTERMAYR CR, 1996, CHEMOMETR INTELL LAB, V34, P187
▲	20	324	WALCZAK B, 1997, CHEMOMETR INTELL LAB, V36, P81
●	21	339	PRESS WH, 1992, NUMERICAL RECIPES C
	22	355	DONOHO DL, 1995, IEEE T INFORM THEORY, V41, P613
●	23	371	SHAPIRO JM, 1993, IEEE T SIGNAL PROCES, V41, P3445
●	24	548	DONOHO DL, 1995, J AM STAT ASSOC, V90, P1200
●	25	559	DAUBECHIES I, 1992, 10 LECT WAVELETS
●	26	595	WITTEN IH, 1987, COMMUN ACM, V30, P520
● ●	27	604	JOSHI RL, 1995, IEEE T CIRC SYST VID, V5, P515
● ●	28	605	JOSHI RL, 1997, IEEE T IMAGE PROCESS, V6, P1473
●	29	611	SAID A, 1996, IEEE T CIRC SYST VID, V6, P243
	30	618	COIFMAN RR, 1992, IEEE T INFORM THEORY, V38, P713
●	31	723	DONOHO DL, 1995, J ROY STAT SOC B MET, V57, P301
	32	742	GROSSMANN A, 1984, SIAM J MATH ANAL, V15, P723
	33	1061	GABOR D, 1946, J I ELEC ENG 3, V93, P429
	34	1387	KAISER G, 1994, FRIENDLY GUIDE WAVEL

I now expand the data set size, to examine the effect of a larger collection on clustering and frequent itemsets. For the same was 1999 "wavelet*" query, I consider the first 150 documents, which cite a total of 3756 documents. After citation-count filtering (retaining documents cited 3 or more times), 53 cited documents remain. The results are shown in Figure 3-26.

In the figure, there are 6 frequent itemsets of cardinality 4. One of these itemsets is the set of 3 chemistry papers plus Mallat's paper that we found for the smaller data set. The 5 remaining cardinality-4 itemsets overlap extensively. In fact, they form a cardinality-5 frequent itemset. Again, these are technology papers in image coding.

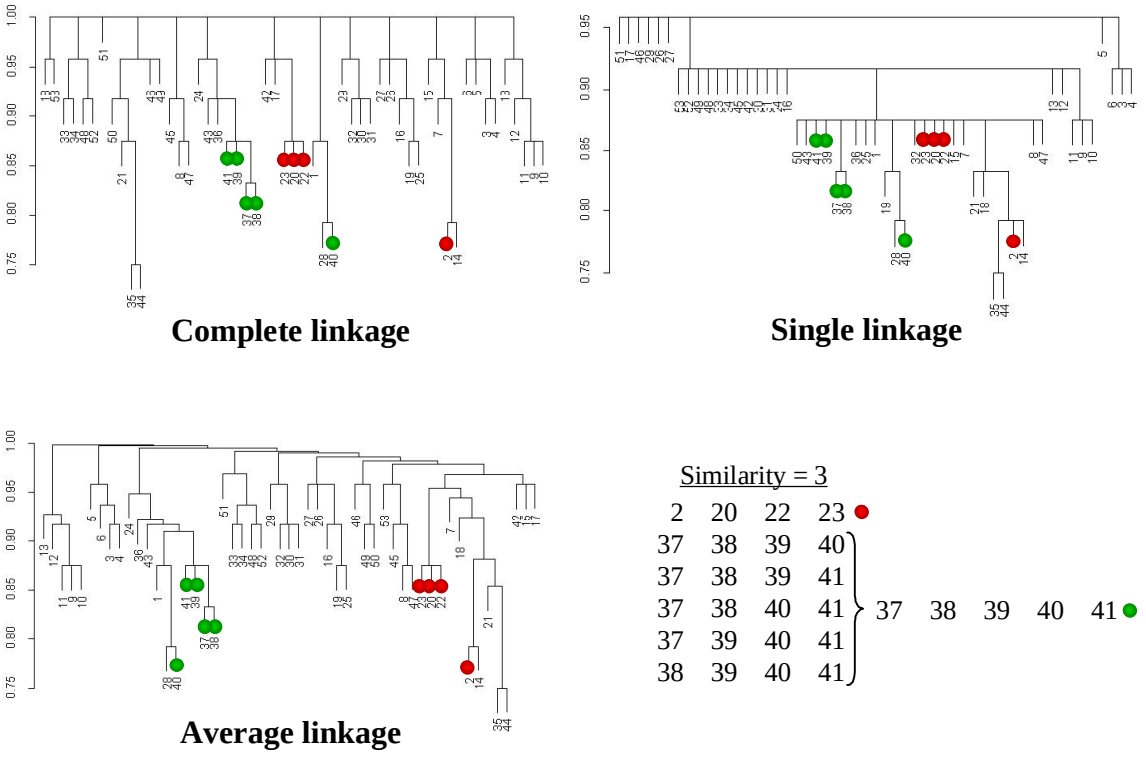


Figure 3-26: Clustering versus frequent itemsets for "Wavelets (1-150)" data set.

For complete-linkage, both the cardinality-4 and the cardinality-5 frequent itemsets form clusters only by including the entire data set in the cluster. The situation is better with single-linkage, in which a cluster of 26 documents is necessary to completely contain either of the itemsets. The match is best for average linkage, in which the cardinality-4 itemset forms a cluster along with 6 other documents, and the cardinality-5 itemset forms a cluster along with 2 other documents. Overall, clustering with mere pairwise distances does a poor job of matching higher-cardinality frequent itemsets.

Figure 3-27 shows the augmentation of the dendrogram with bibliographic details along with the glyphs for frequent itemsets. This is particularly convenient for information retrieval. Text provides the necessary semantic context for the clustering. Moreover, the itemset glyphs allow the discovery of larger sets of frequently co-cited documents that are not apparent in pairwise clustering.

In my comparison of clustering to frequent itemsets, I have been asking whether the itemsets form clusters comprised only of the itemset members. Stated in another way, I have been determining the minimal-cardinality cluster that contains all the members of a given itemset, and comparing that cluster cardinality to the itemset cardinality. This 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.

I will describe this itemset-matching metric more formally. Let $\pi = \{\pi_1, \pi_2, \ldots, \pi_{k_1}\}$ be a partition of items consistent with the hierarchical clustering merge tree. Furthermore, let $I = \{I_1, I_2, \ldots, I_{k_2}\}$ be a set of itemsets. Then for each itemset $I_i \in I$, there is some block of the partition $\pi_j \in \pi$ such that $|\pi_j|$ is minimized,

subject to the constraint that $I_i \subseteq \pi_j$. I call this π_j the minimal cluster containing the itemset.

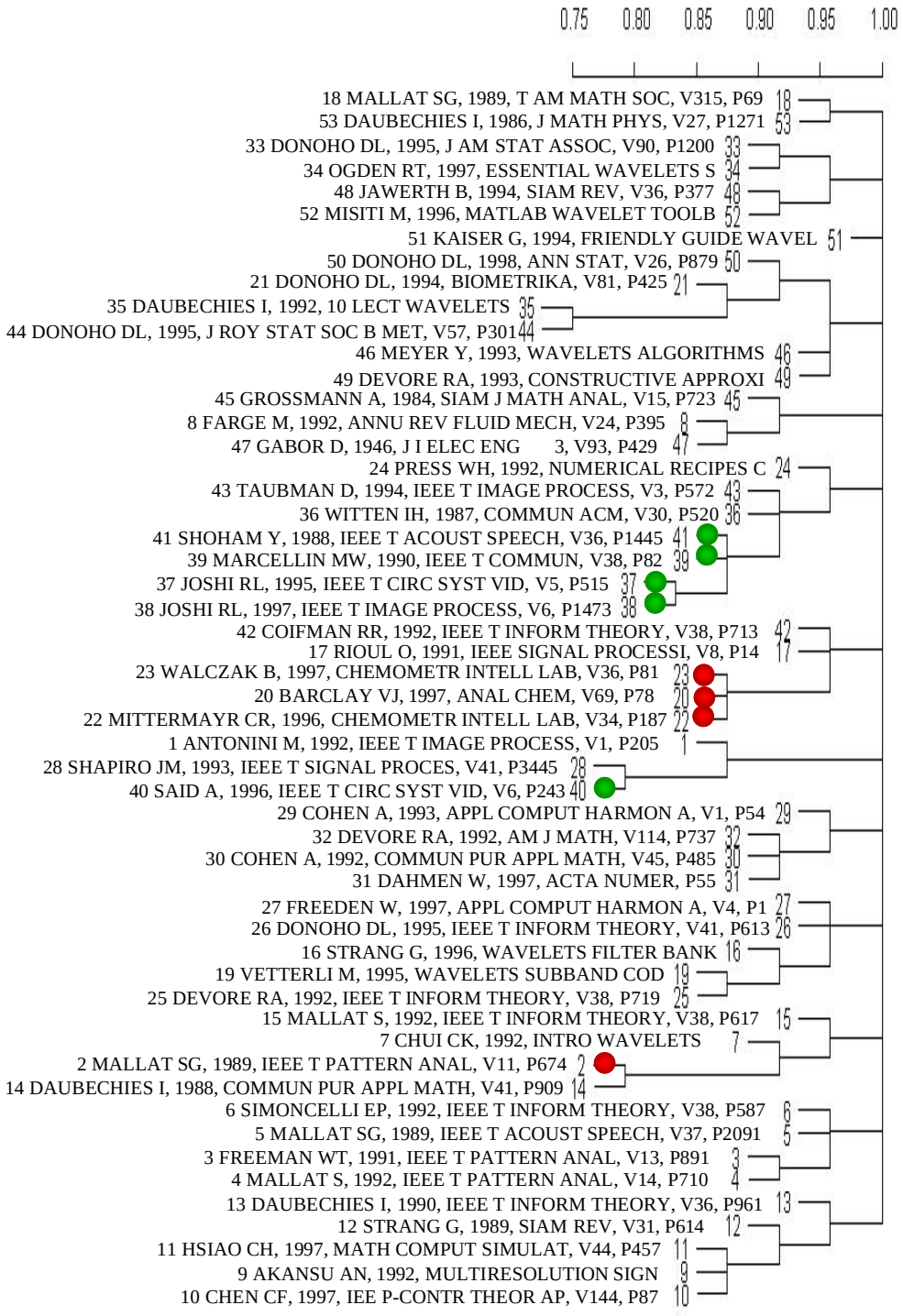


Figure 3-27: Augmenting dendrogram with frequent itemsets and text for information retrieval.

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 $\pi_j \in \pi$ 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,

$$M(\pi, I_i) = \frac{|I_i|}{|\pi_j|}. \tag{3.16}$$

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

$$M(\pi, I) = \frac{1}{|I|} \sum_{I_i \in I} M(\pi, I_i) = \frac{1}{|I|} \sum_{I_i \in I} \left(\frac{|I_i|}{|\pi_j|} \right). \tag{3.17}$$

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 since $|I_i| = |\pi_j|$,

$$M(\pi, I) = \frac{1}{|I|} \sum_{I_i \in I} 1 = \frac{|I|}{|I|} = 1. \tag{3.18}$$

The minimum value of $M(\pi, I)$ is $M(\pi, I) = |I_i|/n$, indicating the poorest possible match. For the proof, consider that $M(\pi, I_i) = |I_i| / |\pi_j|$ 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,

$$M(\pi, I) = \frac{1}{|I|} \sum_{I_i \in I} \left(\frac{|I_i|}{|\pi_j|} \right) = \frac{1}{|I|} \sum_{I_i \in I} \left(\frac{|I_i|}{n} \right) = \frac{1}{|I|} \frac{|I_i|}{n} \sum_{I_i \in I} 1 = \frac{|I|}{|I|} \frac{|I_i|}{n} = \frac{|I_i|}{n}. \tag{3.19}$$

Figure 3-28 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.

The general method of clustering documents and computing itemset-matching clustering metrics is shown in Figure 3-29. As I show in the next section, higher-order co-citations can be included in pairwise document distances. Frequent itemsets of given cardinalities and minimum supports can be also be computed. Once a clustering is

computed for the given distance function, it can be measured by the itemset-matching metric for the given itemsets.

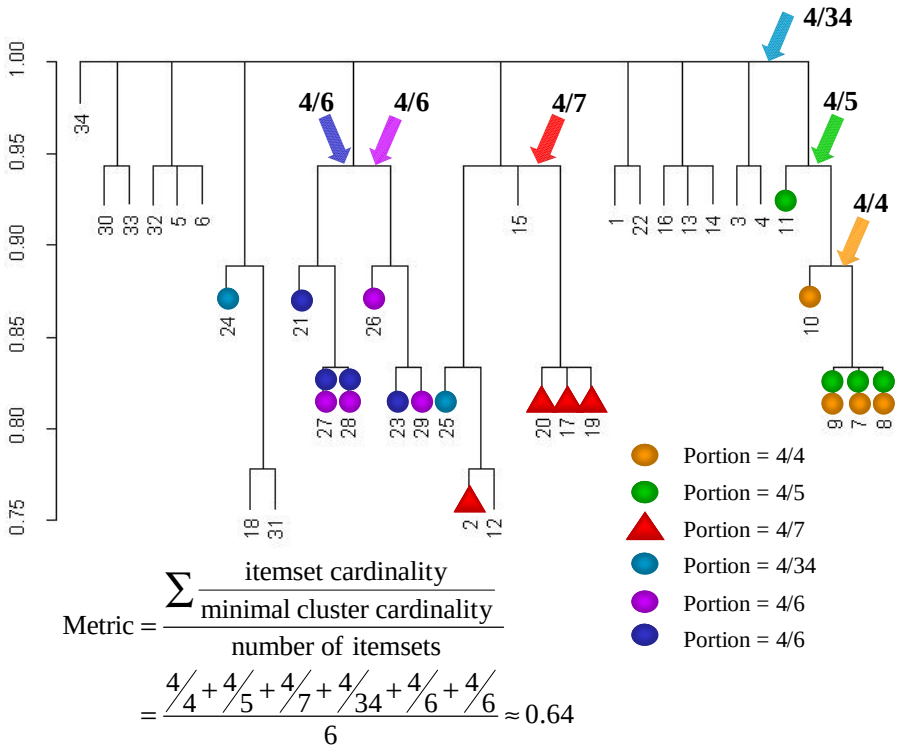


Figure 3-28: Itemset-matching clustering metric is the average portion occupied by an itemset within the minimal cluster containing it.

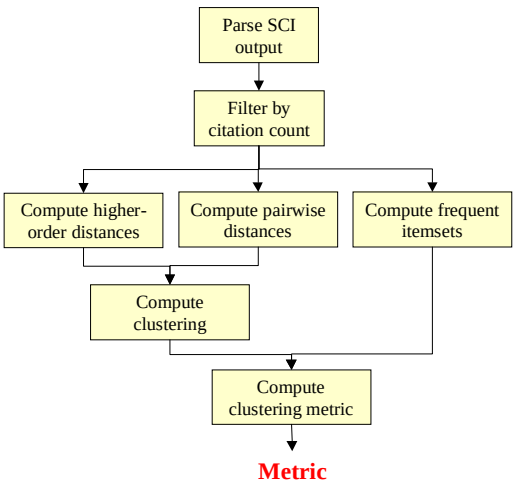


Figure 3-29: General method for computing itemset-matching clustering metric.

3.4 Distances from Higher-Order Co-Citations

A central problem in data mining is the discovery of frequent itemsets. In the context of hypertext systems, such frequent itemsets represent groups of highly similar documents based on higher-order co-citations. But managing and interacting with itemsets for information 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 2 vertices. It is difficult to generalize clustering or minimum spanning tree algorithms to such distance hypergraphs.

My approach is to apply standard clustering or minimum spanning tree algorithms, but with pairwise distances that include higher-order co-citation similarities. The new distances I 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 itemset-matching metric I proposed in the previous section is therefore reasonable.

I now describe the chain of reasoning leading to these new pairwise/higher-order hybrid distances. This reasoning is illustrated by a real-world example with SCI citation

data. The design of the new hybrid distances is guided by performance in terms of the previously proposed itemset-matching metric.

Consider the lattice of itemsets partially ordered by the set inclusion relation, as described in the previous section. Associated with each itemset in the lattice is its support. At the lowest level of the lattice are itemsets of cardinality one. Consider any 2 of these lowest level itemsets, say $\{j\}$ and $\{k\}$. The least upper bound of $\{j\}$ and $\{k\}$ is the cardinality-2 itemset $\{j,k\}$. 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 $d(j,k) = d(k,j)$ between documents j and k 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 $d(j,k)$, as in Garfield's co-citation analysis [Garf79], 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 either multiplicative or additive inve rsion.

A straightforward model for higher-order document similarities is to sum supports over all itemsets that contain the document pair in question. Here the itemset supports can be interpreted as features. More formally, the itemset support feature summation is

$$s_{j,k} = \sum_{\{I \mid j,k \in I\}} \zeta(I). \tag{3.20}$$

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

$$\hat{s}_{j,k} = \frac{s_{j,k} - \min(s_{j,k})}{\max(s_{j,k}) - \min(s_{j,k})}, \tag{3.21}$$

yielding the normalized similarity $\hat{s}_{j,k} \in [0,1]$. I then choose the additive inverse conversion from similarity to distance:

$$d_{j,k} = 1 - \hat{s}_{j,k}. \tag{3.22}$$

I now demonstrate this higher-order distance formula with the "Wavelet (1-100)" data set described in the previous section. The results are shown in Figure 3-30. In terms of matching frequent itemsets, this straightforward model offers no real improvement over simple pairwise similarity.

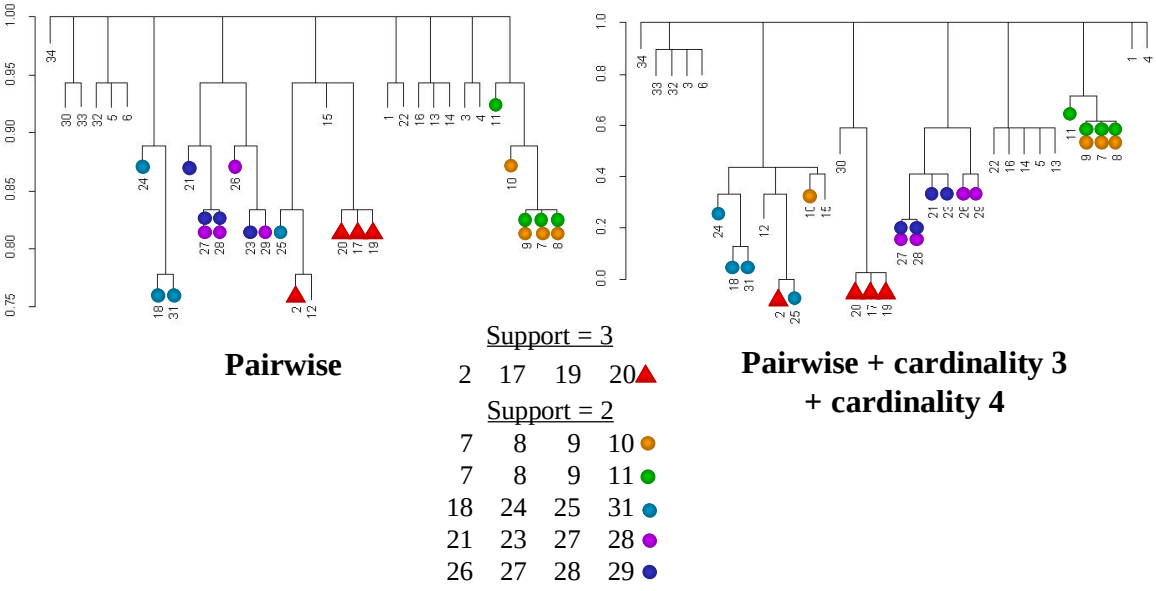


Figure 3-30: Pairwise document similarities by summing supports over all itemsets containing a given document pair.

Rather than sum supports over all itemsets that contain a given document pair, I will sum over itemsets of a given order (cardinality) χ only. This may help us understand the reason for the lack of improvement for our straightforward model. The

document similarities become

$$s_{j,k} = \sum_{\{I \mid j,k \in I, \mid I \mid = \chi\}} \zeta(I). \tag{3.23}$$

Figure 3-31 shows the resulting clusterings for cardinalities $\chi = 2,3,4$. The $\chi = 2$ case is standard pairwise similarity. The $\chi = 3$ case is an improvement over pairwise similarity, with the exception of document #10. However, the $\chi = 4$ case is slightly worse than pairwise similarity.

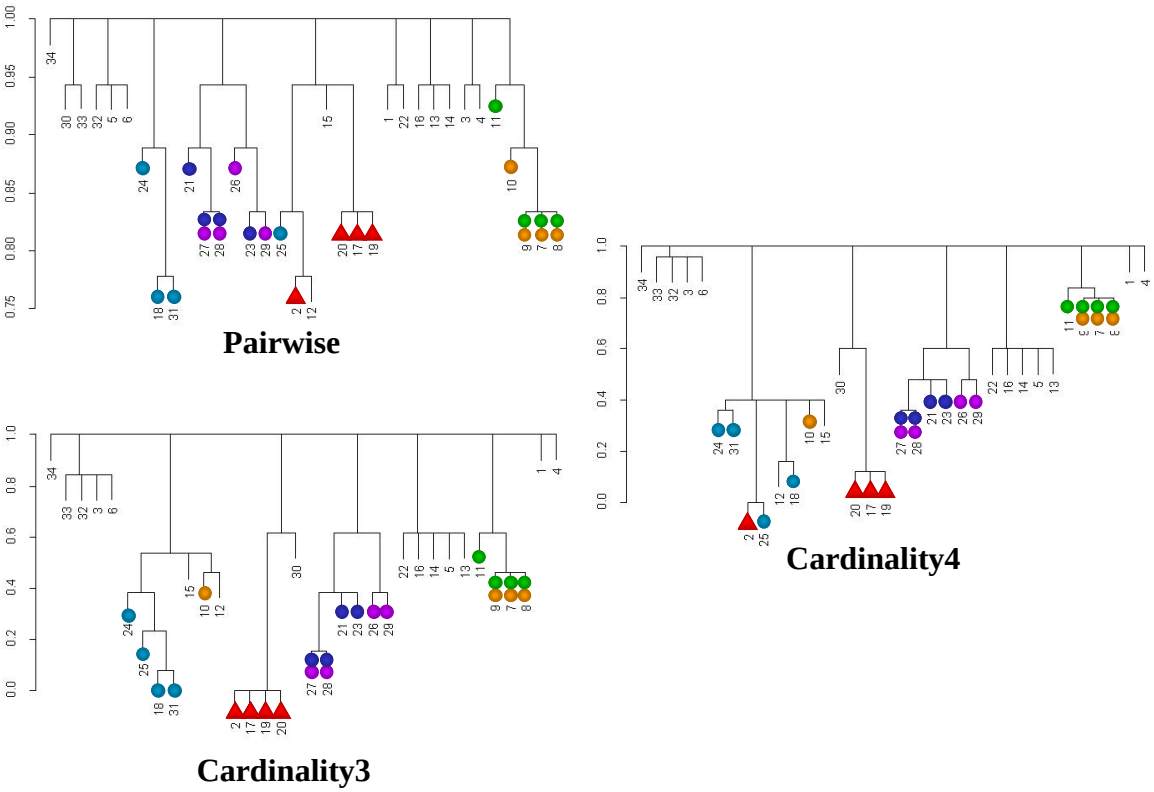


Figure 3-31: Document similarities by summing support features over all itemsets of a given cardinality containing a given document pair.

The poor match of the $\chi = 4$ case helps to clarify things. After all, it sums supports for itemsets of cardinality $\chi = 4$ only, which is the cardinality of the frequent itemsets we are trying to match. This suggests that the supports of the frequent itemsets are insufficiently large to compete with the sum of the supports of the other itemsets.

I propose that a nonlinear transformation $T[\zeta(I)]$ be applied to the itemset supports $\zeta(I)$ before summation. The transformation T should 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 one or an increasing exponential. In general, the nonlinear transformation T could vary with itemset, though here I consider constant T only. The hybrid similarity $s_{j,k}$ then becomes

$$s_{j,k} = \sum_{\{I | j,k \in I, |I| = \chi\}} T[\zeta(I)]. \tag{3.24}$$

Figure 3-32 demonstrates this hybrid similarity for the nonlinearity $T[\zeta(I)] = [\zeta(I)]^2$. Compare this to the non-transformed similarities via in Eq. (3.23) and shown in Figure 3-31. For similarities with itemset cardinality $\chi = 2$, the nonlinear transformation makes no difference in clustering.

But for cardinality $\chi = 3$, the frequent itemset {7, 8, 9, 10} is now a cluster, whereas before the only cluster containing the itemset was the full set of n documents. Because of the overlap of frequent itemsets {21, 23, 27, 28} and {26, 27, 28, 29}, the cluster them exclusively is probably the closest itemset match we can hope for. The same can be said for frequent itemsets {7, 8, 9, 10} and {7, 8, 9, 11}.

For cardinality $\chi = 3$, the most obvious effect of the nonlinear transformation is that the most frequent itemset {2, 17, 19, 20} is now a cluster. There are still problems

for {7, 8, 9, 10}● and {18, 24, 25, 31}●, but overall the nonlinear transformation offers a solid improvement.

My proposed approach of nonlinearly transforming itemset-support features shows promise for yielding similarities that improve the match between clusters and frequent itemsets. In fact, we can eliminate the constraint that only supports for itemsets of a fixed cardinality χ be included in the similarity computation. Indeed, I only added the constraint to help gain insight into the eff ect of the nonlinear transformation in terms of clustering. The similarity computation then becomes

$$s_{j,k} = \sum_{\{I \mid j,k \in I\}} T[\zeta(I)]. \tag{3.25}$$

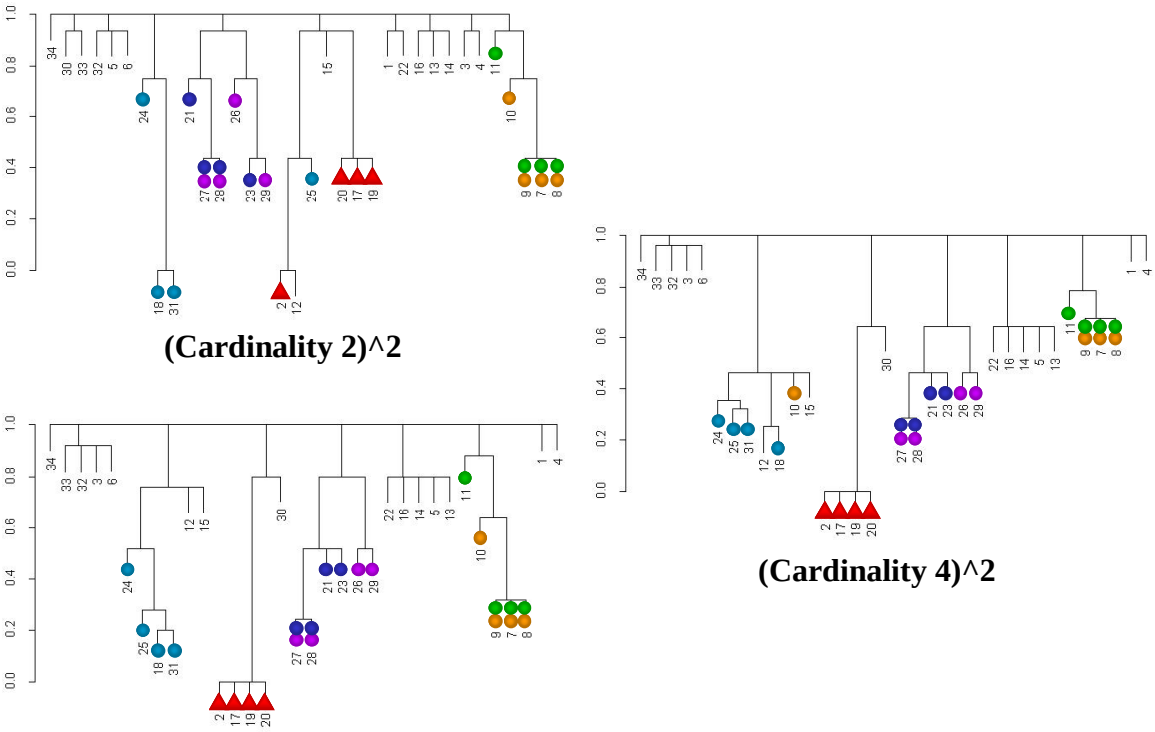


Figure 3-32: Quadratic nonlinearity applied to itemset-support features in computing hybrid pairwise/higher-order similarities.

I now offer a theoretical guarantee for the nonlinear transformation of itemset-support features in the similarity computation. I begin with a proof that the most frequent itemset can always be made a cluster, given a large enough degree of nonlinearity in the transformation. I will then generalize the proof to a comparison of the clustering of arbitrary itemsets in terms of their relative supports.

Consider a given set of itemsets I included in the similarity computation in Eq. (3.25). Then there is some itemset $I_\alpha \in I$ whose support is the largest, i.e. $\zeta(I_\alpha) = \max[\zeta(I)]$.

Consider also a nonlinear transformation $T = T_p = T(\zeta; p)$ that is a function of itemset support ζ, and is parameterized by nonlinearity degree $p > 1$. Transformation T has the property that $T(\zeta_1; p) = \Omega[T(\zeta_2; p)]$ when $\zeta_1 > \zeta_2$. That is, T with a larger support value ζ is asymptotically bounded from below by T with a smaller support value ζ. Example transformations include $T_p(\zeta) = \zeta^P$ and $T_p(\zeta) = p^\zeta$.

Next, there are $C_2^{|I_\alpha|}$ similarities s_{j_1,k_1} whose corresponding document pairs $\{j_1, k_1\}$ are subsets of the maximum-support itemset I_α, that is, $\{s_{j_1,k_1} | \{j_1, k_1\} \subset I_\alpha\}$. These similarities are computed as

$$s_{j_1,k_1} = T_p[\zeta(I_\alpha)] + \sum_{\{I_1 | j_1,k_1 \in I_1, I_\alpha \notin I_1\}} T_p[\zeta(I_1)]. \tag{3.26}$$

The remaining $C_2^n - C_2^{|I_\alpha|}$ similarities $\bar{s}_{j_2,k_2}$ correspond to pairs $\{j_2, k_2\}$ that are *not* subsets of I_α, i.e. $\{s_{j_2,k_2} | \{j_2, k_2\} \not\subset I_\alpha\}$, computed as

$$\bar{s}_{j_2,k_2} = \sum_{\{I_2 | j_2,k_2 \in I_2\}} T_p[\zeta(I_2)]. \tag{3.27}$$

If $s_{j_1,k_1} > \bar{s}_{j_2,k_2} \forall \{j_1,k_1\},\{j_2,k_2\}$, then members of pairs with similarity s_{j_1,k_1} form a cluster exclusive of all other items, since there is some threshold t such that $\min(s_{j_1,k_1}) > t > \max(\bar{s}_{j_2,k_2})$. These exclusively clustered pair members are precisely the members of I_α. Note that the threshold inequality here is in terms of similarities rather than dissimilarities (distances). It continues to hold for any similarity/distance conversion that merely reverses ordering.

For an arbitrary nonlinear parameter p, it could be that $s_{j_1,k_1} < \bar{s}_{j_2,k_2}$ for some $\{j_1,k_1\},\{j_2,k_2\}$, in which case the members of itemset I_α are not exclusively clustered. But from Eq. (3.26),

$$\begin{aligned} s_{j_1,k_1} &= T_p[\zeta(I_\alpha)] + \sum_{\{I_1 | j_1,k_1 \in I_1, I_\alpha \notin I_1\}} T_p[\zeta(I_1)] \\ &= \Theta\{T_p[\zeta(I_\alpha)]\}, \end{aligned} \tag{3.28}$$

under our assumption for T that $\zeta(I_\alpha) > \zeta(I_1)$ implies $T_p[\zeta(I_\alpha)] = \Omega\{T_p[\zeta(I_1)]\}$. Also, from Eq. (3.27),

$$\begin{aligned} \bar{s}_{j_2,k_2} &= \sum_{\{I_2 | I_2 \in I\}} T_p[\zeta(I_2)] \\ &= \Theta\{T_p[\max(\zeta(I_2))]\}, \end{aligned} \tag{3.29}$$

under the same assumption for T. Now, since $\zeta(I_\alpha) > \max[\zeta(I_2)]$,

$$T_p[\zeta(I_\alpha)] = \Omega\{T_p[\max(\zeta(I_2))]\}, \tag{3.30}$$

so that

$$s_{j_1,k_1} = \Omega(\bar{s}_{j_2,k_2}). \tag{3.31}$$

Thus there is always some large enough nonlinearity parameter p such that $s_{j_1,k_1} > \bar{s}_{j_2,k_2} \forall \{j_1,k_1\},\{j_2,k_2\}$, making I_α a cluster. This is what I wished to prove.

This guarantee for clusters matching frequent itemsets can be generalized to itemsets other than the single most frequent one $I_\alpha \in I$. Consider the next most frequent itemset $I_\beta \in I$. More formally, $\zeta(I_\alpha) > \zeta(I_\beta) > \zeta(I_\gamma)$, where $I_\gamma \in I$ represents all other itemsets, i.e. $I_\gamma = I - I_\alpha - I_\beta$. Given the 2 most frequent itemsets I_α and I_β, there are 2 possibilities: either $I_\alpha \cap I_\beta \neq \varnothing$ (they overlap), or else $I_\alpha \cap I_\beta = \varnothing$ (they do not overlap). Here the possibility $I_\alpha \subset I_\beta$ is ruled out, because a property of itemsets implies $\zeta(I_\alpha) \leq \zeta(I_\beta)$, contradicting the assumption that I_α has maximum support.

If itemsets I_α and I_β overlap, then I_α forms a cluster for sufficiently large p, according to the theorem I just proved. This precludes the possibility of I_β asymptotically forming a cluster, since some of its members are already in the asymptotic I_α cluster.

But if itemsets I_α and I_β do *not* overlap, then the condition

$$\min(s_\alpha) > t_1 > \max(s_\beta) \geq \min(s_\beta) > t_2 \geq \max(s_\gamma) \tag{3.32}$$

necessary for both I_α and I_β to be clusters is guaranteed by

$$\begin{aligned} s_\alpha &= \Omega(s_\beta) \\ s_\beta &= \Omega(s_\gamma) \end{aligned} \tag{3.33}$$

Here s_α are the similarities from Eq. (3.26), where the subscripts j_1, k_1 have been replaced for brevity, i.e. $s_\alpha \equiv s_{j_1,k_1}$. The similarities

$$s_\beta \equiv (s_\beta)_{j_3,k_3} = T_p[\zeta(I_\beta)] + \sum_{\{I_3 | j_3,k_3 \in I_3, I_\beta \not\subset I_3\}} T_p[\zeta(I_3)] \tag{3.34}$$

correspond to the $C_2^{|I_\beta|}$ document pairs $\{j_3, k_3\}$ that are subsets of the itemset I_β, that is, $\{(s_\beta)_{j_3,k_3} | \{j_3, k_3\} \subset I_\beta\}$.

The first part of Eq. (3.33) is just Eq. (3.31) in another guise, via $s_{j_1,k_1} \to s_\alpha$ and $\bar{s}_{j_2,k_2} \to s_\beta$, which has already been proven. The proof of the 2nd part of Eq. (3.33), starting from Eq. (3.34), follows the same pattern as the proof of Eq. (3.31) starting from Eq. (3.26), with the key step that $\zeta(I_\beta) > \zeta(I_\gamma)$ implies $T_p[\zeta(I_\beta)] = \Omega\{T_p[\zeta(I_\gamma)]\}$.

This then proceeds by induction to cover all other itemsets. 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 large enough value of the nonlinearity parameter p. The theorem 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.

Perhaps we can gain some empirical insight on necessary degrees of nonlinearity p. Figures 3-33 through 3-35 show clusterings for the SCI "Wavelets (1-100)" data set, with the application of nonlinear transformations $T[\zeta(I)] = [\zeta(I)]^p$. The figures correspond, respectively, to nonlinearities $p = 3,4,5$. Each figure gives clusterings for itemset cardinalities $\chi = 2,3,4$ yielding hybrid pairwise/higher-order distances according to Eq. (3.24). Compare these 3 figures with Figure 3-32, in which a quadratic nonlinearity is applied.

For cardinalities $\chi = 2$, the clusterings are topologically identical regardless of the degree of nonlinearity. The standard pairwise distances are insufficient for matching

higher-cardinality itemsets, even with the application of nonlinearities. The cardinality $\chi = 3$ case agrees with the quadratic nonlinearity case in terms of matching frequent itemsets. It has reached a form of saturation with respect to increasing nonlinearity. For cardinality $\chi = 4$, there is improvement for each new value of nonlinearity p, until it reaches itemset agreement (and nonlinearity saturation) for $p = 4$.

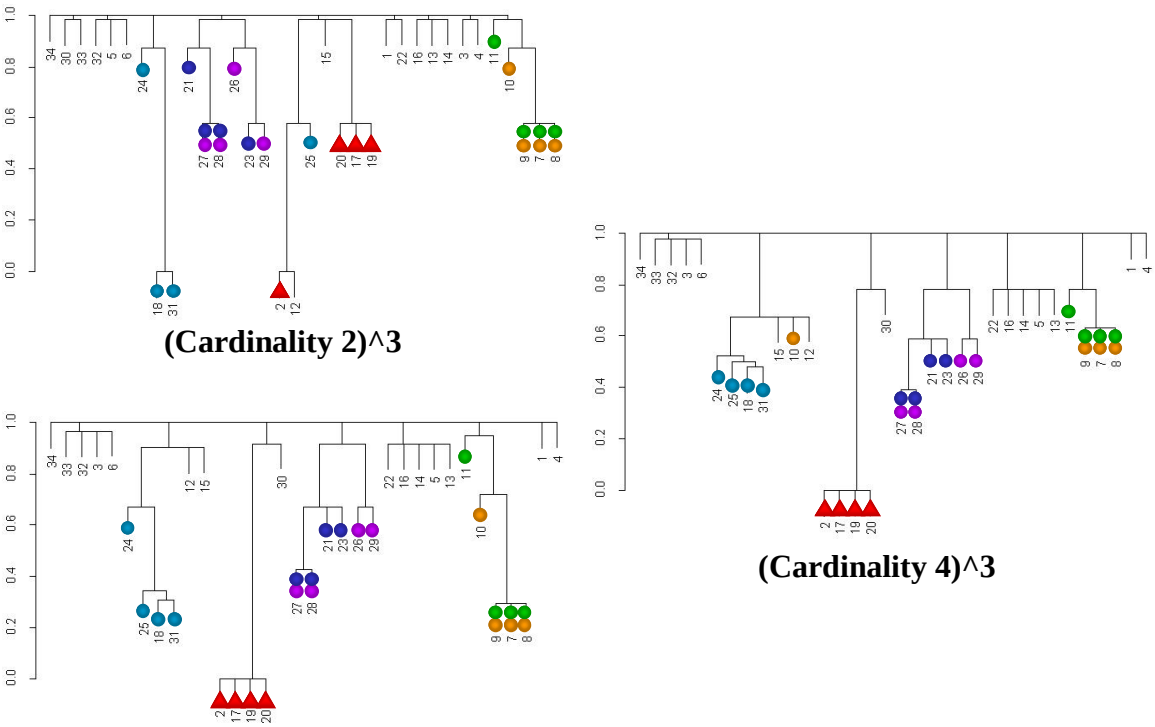
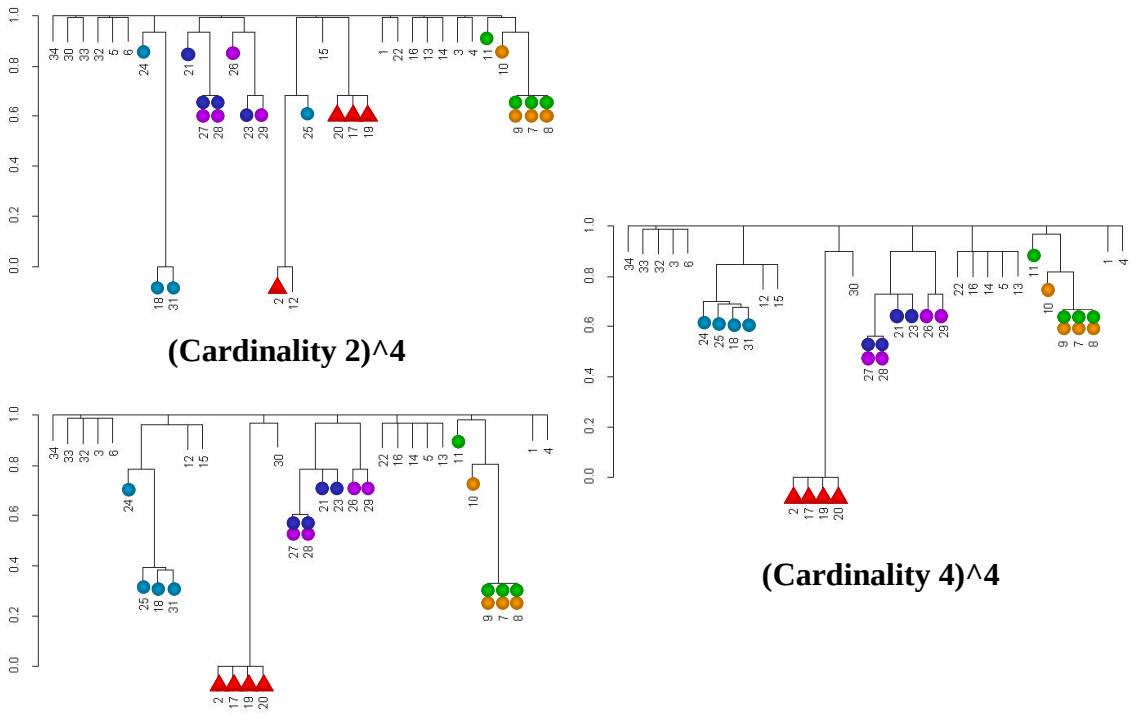


Figure 3-33: Clustering versus frequent itemsets for cubic transformation of itemset-support features.

For this data set, a degree of nonlinearity $p = 4$ in the hybrid pairwise/higher-order distances is sufficient for having clusters match itemsets. In Chapter 5, I show for a variety of data sets that this value of p usually results in good agreement between

clusters and frequent itemsets. For the cases in which the value $p = 4$ yields poor results, the value $p = 6$ is sufficient.

Figure 3-36 computes hybrid pairwise/higher-order similarities by summing over multiple values of itemset cardinality χ. In each case the itemset supports $\zeta(I)$ are nonlinearly transformed by $T[\zeta(I)] = [\zeta(I)]^4$, with similarities computed via Eq. (3.25). I consider 3 separate cases: $\chi = 2,3$; $\chi = 2,3,4$; $\chi = 3,4$.



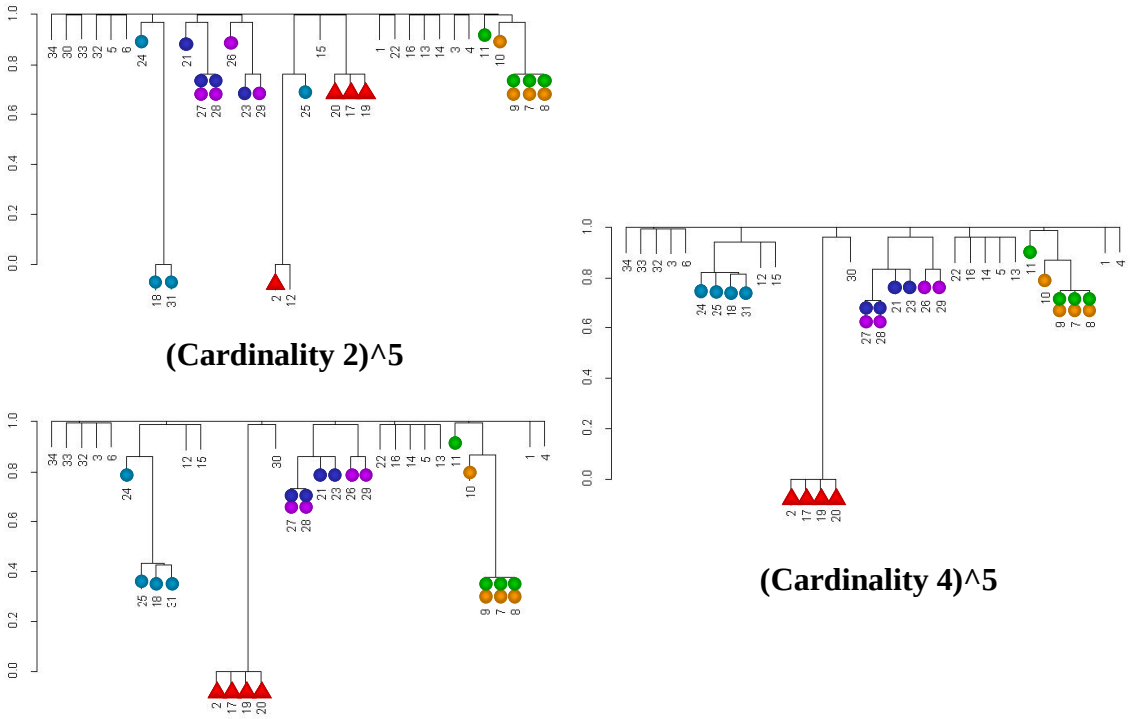
(Cardinality 2)^4

(Cardinality 4)^4

(Cardinality 3)^4

Figure 3-34: Clustering versus frequent itemsets for 4th power transformation of itemset-support features.

There is frequent-itemset agreement for the 2 cases $\chi = 2,3,4$ and $\chi = 3,4$. However, the case $\chi = 2,3$ has disagreement for frequent itemset $\{2,17,19,20\}$▲. It is apparent that the lowest-order (pairwise) supports are the source of the disagreement.



(Cardinality 2)^5

(Cardinality 4)^5

(Cardinality 3)^5

Figure 3-35: Clustering versus frequent itemsets for 5th power transformation of itemset-support features.

In Figure 3-37 I transform itemset supports $\zeta(I)$ exponentially, in particular by $T[\zeta(I)] = 4^{\zeta(I)}$. The transformed supports are summed over single values of itemset cardinality χ. I consider the cases $\chi = 2,3,4$. Clustering results are the same as for the quadratic transformation $T[\zeta(I)] = [\zeta(I)]^2$, shown in Figure 3-32.

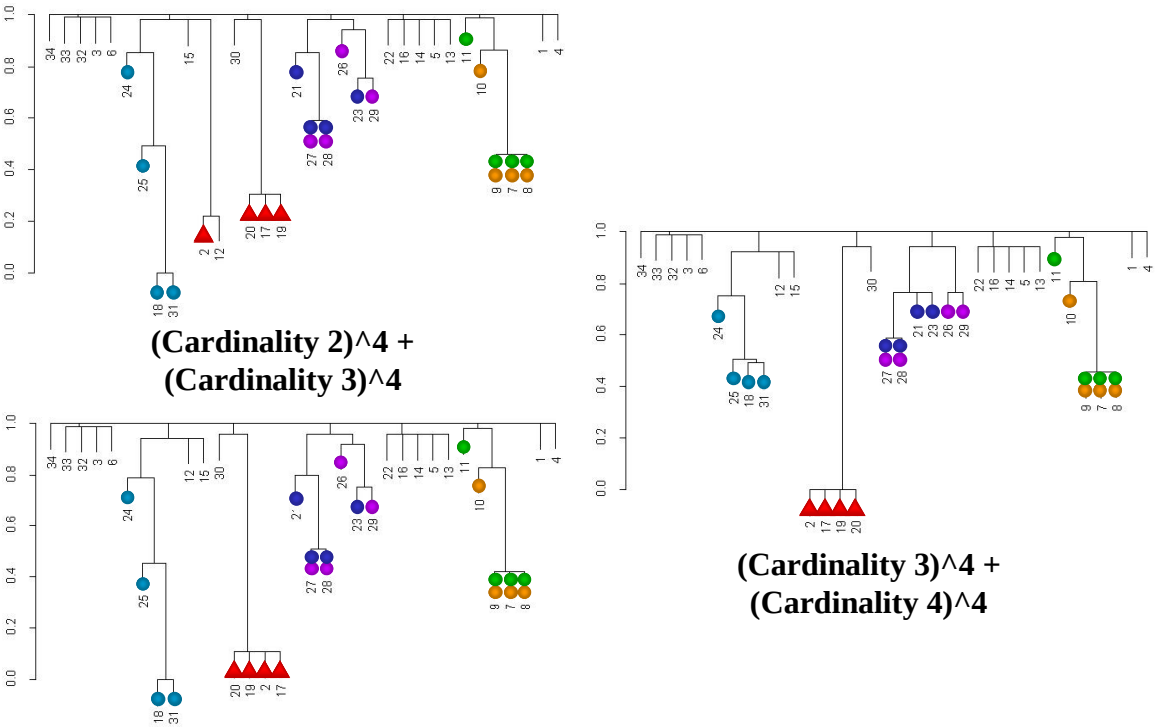


Figure 3-36: Document similarities by summing 4th power supports over itemsets of multiple cardinalities.

In this section, I have proposed a new class of co-citation based inter-document distances. These are a hybrid between pairwise distances and higher-order distances. The new hybrid distances retain a simple pairwise structure, but are better able to match higher cardinality itemsets than are standard pairwise distances. The next section applies these hybrid distances to various sets of documents extracted from the SCI citation database.

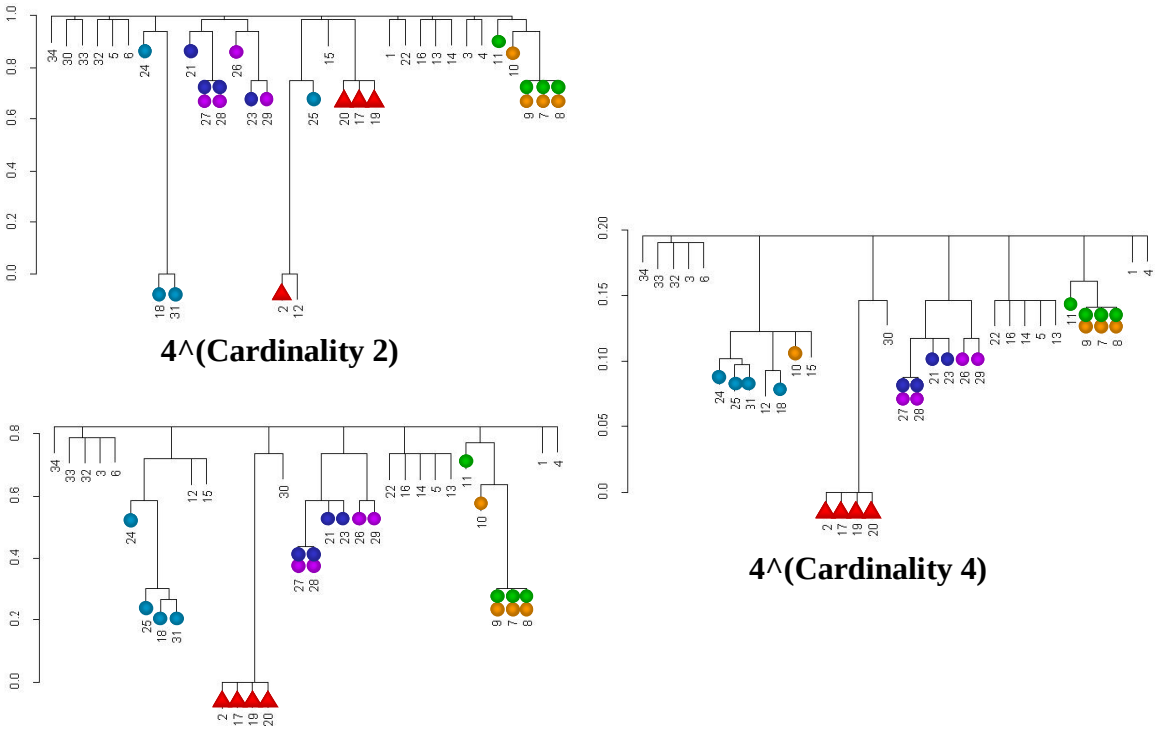


Figure 3-37: Exponential nonlinearity applied to itemset-support features.

3.5 Clustering Experiments

So far in this chapter, I have proposed new methods of visualizing hypertext document clusters for information retrieval. The methodology employs a new class of inter-document distances that is a hybrid between standard pairwise and higher-order distances. The new distances are more consistent with association mining frequent itemsets, and retain a simple pairwise structure.

This section applies the proposed approach to real-world hypertext. In particular, I apply it to data sets from the Institute for Scientific Information's (ISI) Science Citation Index (SCI). Science citations are a classical form of hypertext, and are of interest to our research sponsor [OSTI00]. The SCI data sets employed in this section are described in

Table 3-2. For the data sets "Adaptive Optics," "Quantum Gravity and Strings," and "Wavelets and Brownian," results are included for both co-citations and bibliographic coupling, yielding a total of 10 SCI data sets.

Table 3-2: Details for SCI (Science Citation Index) data sets used in this section. Bibliographic coupling is applied in addition to co-citations for 3 of these data sets.

Data set name	Query keyword	Year(s)	Citing docs	Cited docs
Adaptive Optics	adaptive optics	2000	89	60
Collagen	collagen	1975	494	53
Genetic Algorithms and Neural Networks	genetic algorithm* and neural network*	2000	136	57
Quantum Gravity and Strings	quantum gravity AND string*	1999-2000	114	50
Wavelets (1-100)	wavelet*	1999	100	34
Wavelets (1-500)	wavelet*	1999	472	54
Wavelets and Brownian	wavelet* AND brownian	1973-2000	99	59

Figure 3-38 shows the general method of computing distances for SCI documents, clustering the documents, and comparing clusterings to frequent itemsets via the metric I described in Section 3.3. For the hybrid distances, itemset supports are nonlinearly transformed so that larger supports are relatively exaggerated. Distances may be computed over any number of itemset cardinalities. Likewise, frequent itemsets used in the itemset-comparison metric may have any number of cardinalities, with a given value of minimum support. All of these elements – support nonlinearity, cardinality (both for distances and frequent itemsets), and min-support – significantly affect the value of the clustering metric. As a comparison, I compute metrics for clustering with standard pairwise distances.

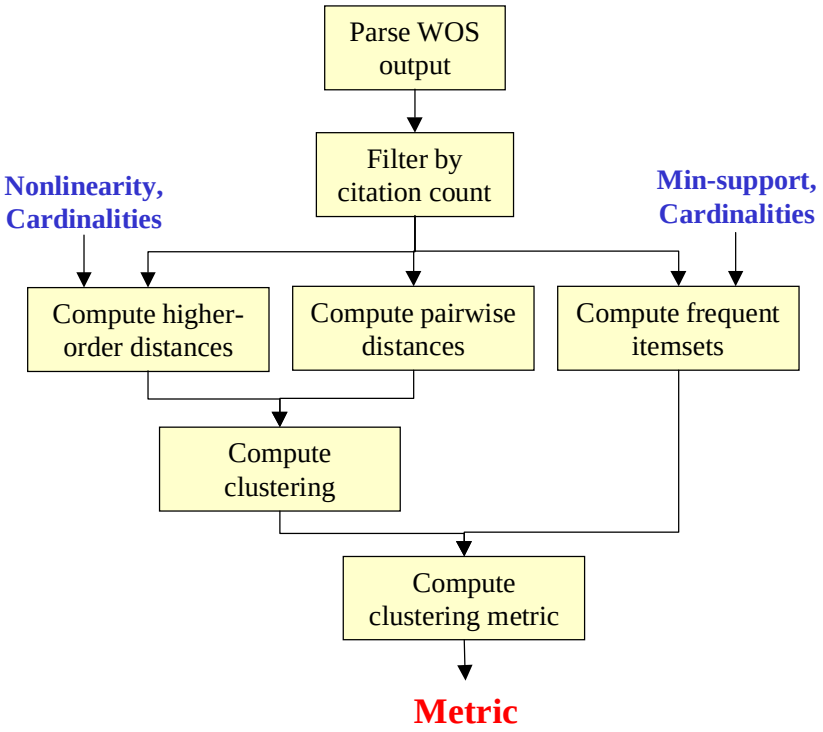


Figure 3-38: General method for computing itemset-matching clustering metric.

Section 3.3 proposed a metric for measuring how well hierarchical clusterings agree with frequent itemsets. The itemset-matching clustering metric $M(\pi, I)$ is

$$M(\pi, I) = \frac{1}{|I|} \sum_{I_i \in I} \left(\frac{|I_i|}{|\pi_j|} \right). \tag{3.35}$$

It is defined for a set of itemsets I and a clustering (partition of the items) π, where $I_i \in I$ is a single itemset, and $\pi_1 \in \pi$ is the minimal cardinality cluster (block of the partition) that contains all items in I_i. Here π is any partition consistent with the hierarchical clustering merge tree. The maximum value of the metric $M(\pi, I)$ is unity, indicating the best possible match between itemsets and clusters, and the minimum value is $M(\pi, I) = |I_i|/n$, indicating the poorest possible match.

The document similarities $s_{j,k}$ forming the basis for standard pairwise distances are simply co-citation counts:

$$s_{j,k} = \sum_i a_{i,j} a_{i,k} \ . \tag{3.36}$$

Here $a_{i,j}$ and $a_{i,k}$ are elements of the citation (hypertext linkage) adjacency matrix, where i indexes citing (linking) documents and j and k index cited (linked) documents.

Hybrid similarities $s_{j,k}$ are computed from higher-order co-citations, equivalent to association mining itemsets, as

$$s_{j,k} = \sum_{\{I | j,k \in I, |I| = \chi\}} T[\zeta(I)]. \tag{3.37}$$

The similarity $s_{j,k}$ between cited documents j and k is a summation over all itemsets I that contain them, for itemset cardinality χ and itemset support $\zeta(I)$. Before summation, the supports are super-linearly transformed by $T = T_p$, which has nonlinearity degree $p > 1$.

Similarities of both types are normalized via

$$\hat{s}_{j,k} = \frac{s_{j,k} - \min(s_{j,k})}{\max(s_{j,k}) - \min(s_{j,k})}, \tag{3.38}$$

and converted to dissimilarities (distances) via linear inversion:

$$d_{j,k} = 1 - \hat{s}_{j,k} . \tag{3.39}$$

Tables A-1 through A-10 in Appendix A show clustering metric results for the SCI data sets. Hierarchical clustering is done via complete-linkage, average-linkage, or single-linkage criterion. Distances are denoted either pairwise or of the form o_χ^p, indicating the summation of itemset supports raised to the power p, over itemsets of

cardinality χ. In comparing standard and hybrid distances, it is important that the same itemsets are applied in computing metric values, and identical clustering criteria are applied. In the tables, such comparable values are demarcated with alternating red and black text.

For a given test case, the k most frequent itemsets are included in the metric, with metric itemset cardinality χ, which is denoted o_χ. In each case, the cardinality χ is consistent between distances o_χ^p and frequent itemsets o_χ (except for pairwise distances, i.e. o_2^1). While such distance/metric cardinality consistency is not required, it allows a more direct interpretation of the metric.

I now provide my interpretation of the clustering metric results in Tables A-1 through A-10. I begin by comparing standard pairwise distances computed from Eq. (3.36) to the hybrid distances computed from Eq. (3.37). The discussion proceeds according to classes of test cases exhibiting similar behavior, as opposed to the purely alphabetical order of the tables.

I begin with the "Wavelets (1-100)" data set that has been featured prominently in previous chapters. Table A-7 shows clustering metrics for this data set. Clusterings from hybrid pairwise/higher-order distances are more consistent with frequent itemsets than are ones from standard pairwise distances, or at least equally as consistent. The metric values generally decrease as the numbers of frequent itemsets in the metric increase. This is because the itemset support nonlinearity favors supports of only the sparse numbers of more frequent itemsets.

Tables A-3, A-5, A-8, and A-9 show clustering metrics for the "Collagen," "Quantum Gravity and Strings," "Wavelets (1-500)," and "Wavelets and Brownian" data

sets, respectively. Results for these 4 data sets are similar to those for the first data set. For almost every pair of cases, itemset-matching clustering metrics for hybrid distances are equal to or greater than those for pairwise similarities. In particular, for these 4 data sets, metric values are higher for pairwise similarities in only 3 out of 144 cases. The average metric value difference for these 3 cases is less than 3 %.

There are 2 data sets in which itemset-matching clustering metrics are not as consistently high for hybrid pairwise/higher-order similarities compared to standard pairwise similarities. These are the "Adaptive Optics" and "Genetic Algorithms and Neural Networks" data sets. Results for these data sets are shown in Tables A-1 and A-4, respectively.

Note the cases in Table A-4 for frequent itemsets $o_3, k = 1$. Clustering metrics for both average-linkage and single-linkage are lower for o_3^4 distances than for pairwise distances. This is especially surprising since only the single most frequent itemset is included in the metric. I decided to increase the itemset support nonlinearity to o_3^6, to increase the exaggeration of large itemsets. Such an increase is in the spirit of the theoretical guarantee I provided in Chapter 3 for frequent itemset matching. The resulting metrics are all unity for frequent itemsets $o_3, k = 1$. In fact, clustering metrics increase for all cases with o_3^6 distances in comparison to o_3^4 distances. For o_4^4 distances, clustering metrics are higher than for pairwise distances.

For the data set in Table A-1, metric values for frequent itemsets defined by $o_3, k = 1$ are all unity. However, for $o_3, k = 5$, metric values for o_3^4 distances are not significantly better than for pairwise distances. Increasing itemset support nonlinearity to

o_3^6 offers no real increase in metric value. The situation is similar for o_4 frequent itemsets, except there are relatively low metric values for higher-order similarities even for the $k = 1$ cases.

The relatively poor match for the data set in Table A-1 can be understood by an examination of the itemset supports. Figure 3-39 shows itemset supports for the data set in Table A-1, with supports for the data set in Table A-8 as a comparison. For the data set in Table A-1, there are relatively many itemsets at the largest support value. When such itemsets have overlapping items, it becomes more difficult to isolate individual itemsets in clusters. In contrast, for the data set in Table A-8, the itemsets are relatively sparse at the larger supports. As the metric values in Table A-8 show, the frequent itemsets are better isolated in clusters.

So far in this section, clustering has been based on co-citations, i.e. the clustering of cit*ed* documents. I now compute metrics for the clustering of cit*ing* documents, known as bibliographic coupling. Mathematically, this is equivalent to a transposition of the citation adjacency matrix A, i.e. $A \rightarrow A^T$. Table A-10 shows clustering metrics for bibliographic coupling of the "Wavelets and Brownian" data set. In each case, metric values are larger for hybrid pairwise/higher-order distances than for pairwise distances.

For 2 other data sets ("Adaptive Optics" and "Quantum Gravity and Strings"), clustering metric results for bibliographic coupling are not as consistent with frequent itemsets. The metric values are in Tables A-6 and A-2.

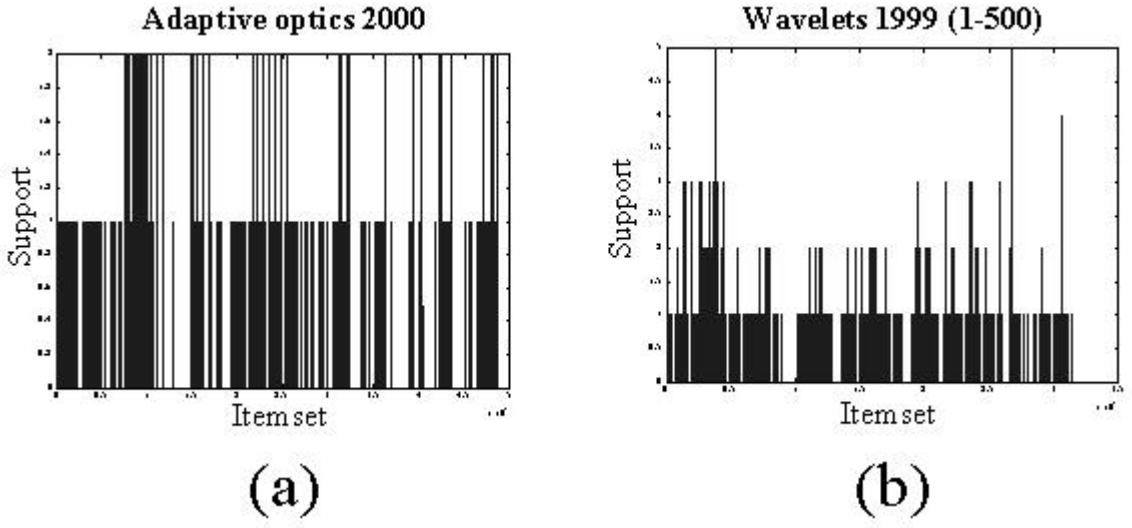


Figure 3-39: Cardinality-4 itemset supports for (a) the data set in Table A-1 and (b) the data set in Table A-8. In (a), there are relatively many itemsets at the largest support value, while in (b) there are relatively few. Multiple overlapping itemsets with larger supports make the isolation of individual itemsets in clusters more difficult.

In Table A-6, for o_3 frequent itemsets, the metrics for hybrid distances are all equal to or larger than pairwise similarities. For o_4, distances are not all smaller than pairwise for o_4^4 hybrid distances. But increasing the support nonlinearity to o_4^6 either raises or causes no change to the metric. In particular, the metrics for o_4^6 are all larger than for pairwise. This is another example of an increase in support nonlinearity that results in better consistency with frequent itemsets.

In Table A-2, for o_3 frequent itemsets, the metrics for hybrid distances are equal to or greater than pairwise distances, as for Table A-6. For $o_4, k=1$ frequent itemsets, hybrid distances are smaller than pairwise for o_4^4 distances, but equal to pairwise distances (both are unity) for o_4^6. That is, raising the support nonlinearity raises the

metric. However, for $o_4, k = 5$ and $o_4, k = 10$, increasing support nonlinearity lowers the metric about as often as it raises it. These metric results are consistent with the itemset supports for the data set, shown in Figure 3-40.

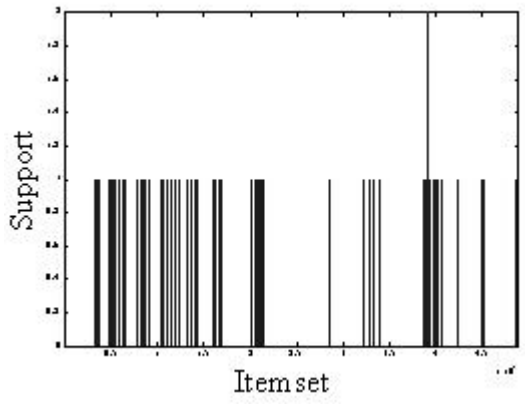


Figure 3-40: Cardinality-4 itemset supports for the data set in Table A-2. There are many itemsets at support = 1, which reduce the clustering metric when there is more than one frequent itemset.

Table 3-3 compares clustering metric values for the test cases in Tables A-1 through A-10. It classifies cases as metric values for hybrid distances being either equal to, greater than, or less than values for standard pairwise distances. Higher metric values correspond to clusters being more consistent with frequent itemsets. For most test cases, the new hybrid pairwise/higher-order distances result in better consistency with frequent itemsets in comparison to standard pairwise distances.

Tables A-1 through A-10 show that itemset support nonlinearities of power 4 are generally sufficient for a good match to frequent itemsets. Otherwise nonlinearities of power 6 are sufficient, in all but one comparison with standard pairwise distances. Here I 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.

Table 3-3: Clustering metric comparisons for standard pairwise (P.W.) vs. hybrid pairwise/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
Totals	29	169	27	225

This section concludes Chapter 3. In this chapter, I have introduced hypertext document distances based on the co-citation relationship. I showed how these distances can be applied to hierarchical clustering, and visualized with the clustering dendrogram. I introduced a new augmentation of the dendrogram with glyphs for frequent itemset members, and proposed a new metric for comparing clusterings to frequent itemsets.

These innovations guided the design of new hybrid pairwise/higher-order distances. The new distances are directly applied to clustering dendrograms, providing a information retrieval user interface that is less complex than dealing with the frequent itemsets directly. I showed that for a large number of test cases, these distances are more consistent with frequent itemsets than are standard pairwise distances. I also provided a

theoretical guarantee for the new distances in terms of their ability to cluster frequent itemsets.

In the next chapter, I describe methods for reducing the computational complexity of these new hybrid distances.

Chapter 4

Reducing Computational Complexity

In Chapter 3, I proposed including higher-order similarities in distance computations, yielding hybrid pairwise/higher-order distances. I then computed standard hierarchical clustering dendrograms with these new hybrid distances. This greatly reduces the complexity of the user interface for information retrieval, while still allowing the exploration of itemsets of cardinality beyond pairwise.

However, recall that there is a combinatorial explosion of numbers of itemsets for increasing numbers of documents and itemset cardinalities, the number of itemsets increasing as $C_{\chi}^{n} = \frac{n!}{k(n-\chi)!}$ for n documents and itemset cardinality χ. The method I proposed in Chapter 3 for computing hybrid pairwise/higher-order distances includes all these combinatorially exploding numbers of itemsets.

This chapter therefore investigates methods for computing these hybrid distances that eliminate some itemsets, thereby reducing computational complexity. Because some itemset supports are missing, the resulting distances are forms of approximation that potentially have poorer matches with frequent itemsets.

I investigate 2 general approaches for reducing the complexity of computing document distances. The first is to compute hybrid pairwise/higher-order distances from frequent itemset supports only, which allows the application of fast algorithms for computing frequent itemsets. This is covered in the next section. Section 4.2 offers

experimental results that are consistent with previous observations that these fast algorithms empirically scale linearly with problem size.

Section 4.3 describes the second approach to reducing computational complexity. This approach computes distances from a citation adjacency matrix whose rows and columns are weighted by numbers of cited and citing documents, respectively. It includes supports for itemsets of cardinality 2 only, thus avoiding the worst-case exponential complexity of higher cardinality itemsets (higher-order co-citations).

4.1 Fast Algorithms for Frequent Itemsets

Perhaps the most obvious approach in reducing computational complexity for hybrid pairwise/higher-order distances is to exclude itemsets whose supports fall below some threshold value, denoted *minsup*. This approach of computing only the more frequent itemsets is typical, and has been studied in the association mining literature [Agra93][Agra94].

Algorithms for computing frequent itemsets are generally based on 2 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 database tuples, 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 algorithms have been proposed that empirically scale linearly with respect to both the number of transactions and the transaction size [Agra94]. In the context of hypertext

systems, transactions are documents linking to other ones. In fact, these fast algorithms for computing frequent-itemsets actually empirically speed up slightly for increasing numbers of items (cited documents).

Recall that with the proposed method for computing hybrid pairwise/higher-order similarities, I first apply a nonlinear transformation $T[\zeta(I)]$ to supports $\zeta(I)$ of itemsets I. The transformation *T* is to asymptotically increase more quickly than linearly, so that transformed larger supports are bounded from below by transformed smaller supports. For similarity $s_{j,k}$ between documents j and k, $j \neq k$, the transformation is followed by summing transformed supports over all itemsets that contain the document pair:

$$s_{j,k} = \sum_{j,k \in I} T[\zeta(I)]. \tag{4.1}$$

It is then straightforward to modify this so as to exclude less frequent itemsets for reducing computational complexity. Simply exclude from the summation in Eq. (4.1) all itemsets with supports $\zeta(I)$ below *minsup*. The similarity $s_{j,k}$ then becomes

$$s_{j,k} = \sum_{\substack{j,k \in I, \\ \zeta(I) \geq minsup}} T[\zeta(I)]. \tag{4.2}$$

As before, similarities are normalized to $[0,1]$ via Eq. (3.21), then dissimilarities (distances) are computed according to Eq. (3.22).

Figure 4-1 shows example itemset supports, both before and after nonlinear transformation, for various values of minimum support *minsup*. Supports are for cardinality-3 itemsets of the SCI "Microtubules" data set. As *minsup* increases, larger supports become increasingly sparse. Per Eq. (4.2), the only transformed supports $T[\zeta(I)]$ included in hybrid pairwise/higher-order obey $\zeta(I) \geq minsup$.

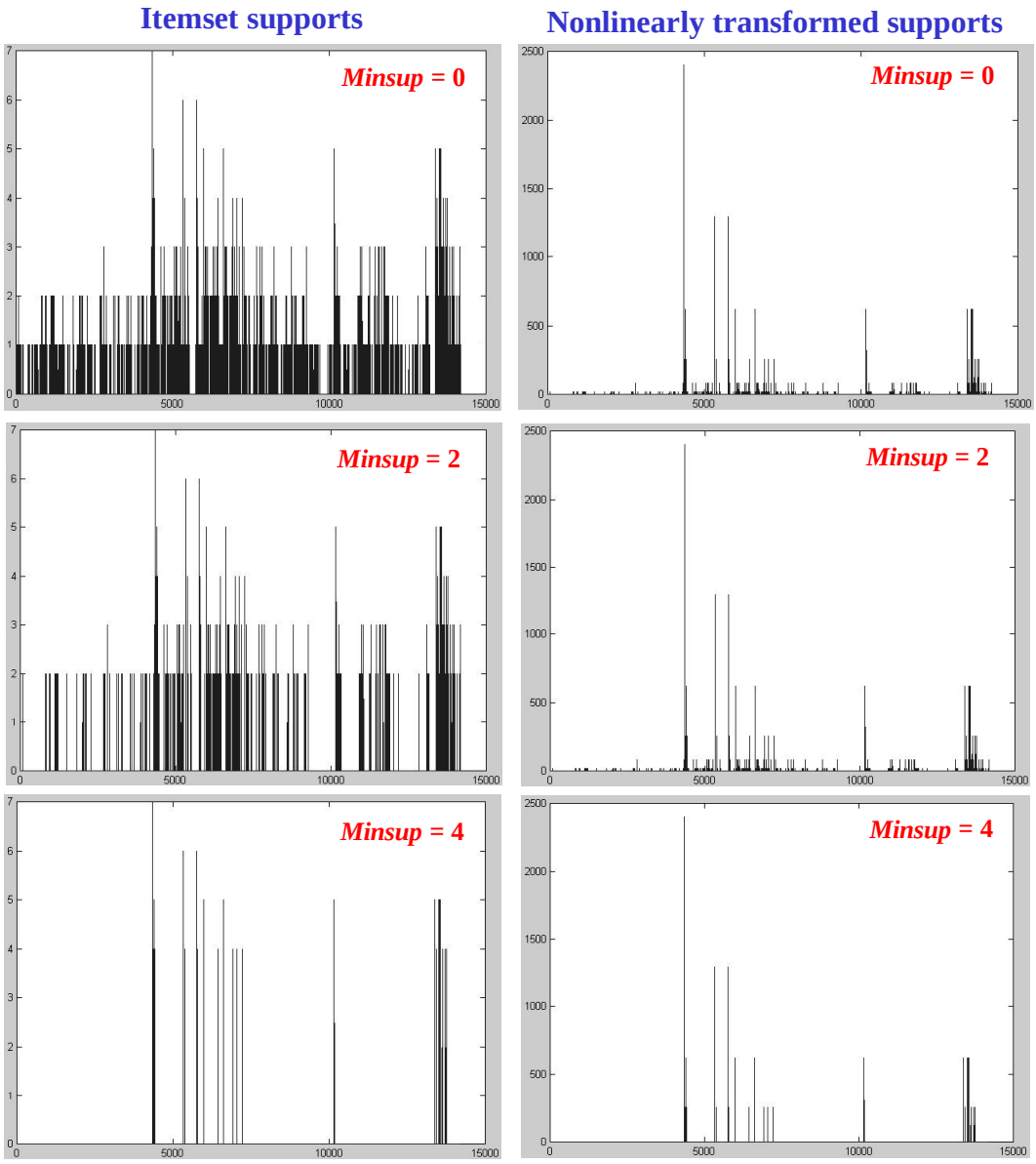


Figure 4-1: Original and nonlinearly transformed itemset supports for 3 different values of *minsup*. Only nonzero transformed supports are included in hybrid distances.

Tables B-1 through B-5 in Appendix B show clustering metrics for the complexity reduction technique described by Eq. (4.2). The metrics are computed for the SCI data sets described in Table 4-1. For the data set "Wavelets and Brownian," results are included for both co-citations and bibliographic coupling, yielding a total of 5 data sets.

Table 4-1: Details for SCI (Science Citation Index) data sets used in this section. Bibliographic coupling is applied in addition to co-citations for one of these data sets.

Data set name	Query keyword	Year(s)	Citing docs	Cited docs
Collagen	collagen	1975	494	53
Quantum Gravity and Strings	quantum gravity AND string*	1999-2000	114	50
Wavelets (1-500)	wavelet*	1999	472	54
Wavelets and Brownian	wavelet* AND brownian	1973-2000	99	59

The clustering metric results in Tables B-1 through B-5 are summarized in Tables 4-2 and 4-3. The results show that excluding itemset supports below *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.

I 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 hybrid distances.

Table 4-2: Clustering metric comparisons for hybrid distances from Chapter 3 (*minsup* 0) versus hybrid distances with reduced complexity (*minsup* 2).

Data set	(*minsup* 2) = (*minsup* 0)	(*minsup* 2) > (*minsup* 0)	(*minsup* 2) < (*minsup* 0)	Cases
1	18	0	0	18
2	18	0	0	18
3	18	0	0	18
4	18	0	0	18
5	11	0	7	18
Totals	83	0	7	90

Table 4-3: Clustering metric comparisons for hybrid distances from Chapter 3 (*minsup* 0) versus hybrid distances with reduced complexity (*minsup* 4).

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

4.2 Itemset Support Distributions

This section offers results that are consistent with the idea that fast algorithms for computing frequent itemsets scale linearly with problem size. In particular, I show for the first time that citation itemset supports generally follow a one-sided Laplacian

(exponentially decreasing) distribution. This distribution also suggests that it should generally be easy for the largest citation itemsets to form clusters via my proposed hybrid pairwise/higher-order distances.

Figure 4-2 illustrates this for the cardinality $\chi = 4$ itemsets of the SCI "Wavelets (1-100)" data set. The figure shows supports for each itemset, along with a log-scale count of the number of itemsets with a given support. The figure also includes the positive side of the Laplacian distribution, which appears linear on the log scale. The general Laplacian distribution $p(x)$ is

$$p(x) = \frac{\lambda}{2} e^{-\lambda|x|}. \tag{4.3}$$

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

$$p^{+}(x) = \lambda e^{-\lambda x}, \quad x \geq 0. \tag{4.4}$$

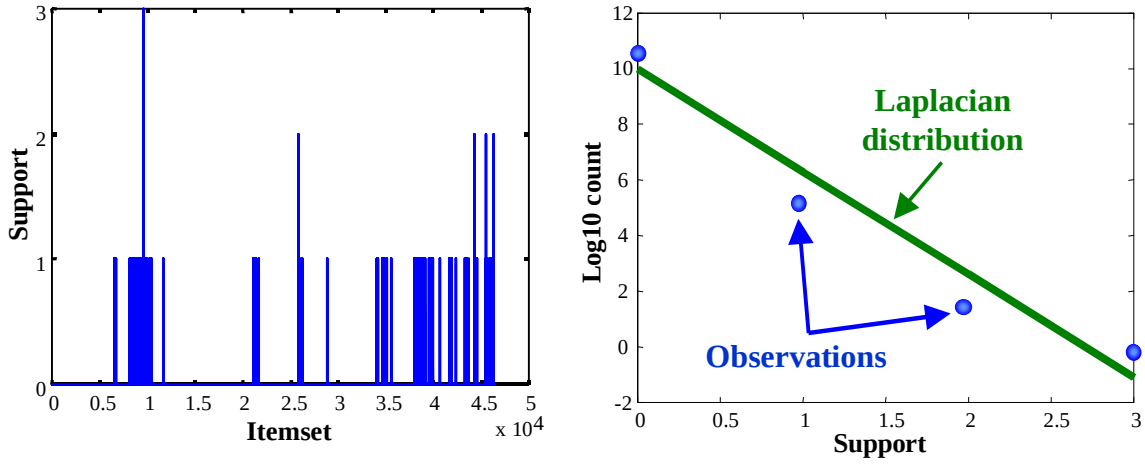


Figure 4-2: Cardinality-4 itemset-support distribution versus right-sided Laplacian for SCI "Wavelets (1-100)" data set.

In terms of the Laplacian distribution's effect on average computational complexity, consider that algorithms generally compute itemsets starting from 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 small supports in the Laplacian distribution, a large proportion of itemsets are below *minsup*, and get removed from any further consideration at higher cardinalities. This in turn contributes to low average computational complexity.

The Laplacian distribution also contributes to the ability of my proposed hybrid pairwise/higher-order distances to produce clusters consistent with frequent itemsets. For example, of the 46376 total itemsets in Figure 4-2, there is only a single itemset having the largest support. According to the theorem I proved in Chapter 3, this itemset is guaranteed to be a cluster, given a large enough degree of nonlinearity.

For the next largest support, there are only 5 itemsets. If any of these have no overlap with either the most frequent itemset or the other 4 itemsets of the same support, they are also guaranteed to be cluster. In general, the sparse nature of more frequent itemsets under the Laplacian distribution improves their chances of being clusters.

Figure 4-3 shows the itemset-support distribution for a data set in which there is a larger range of supports. This data set was generated from the SCI keyword query "microtubules." Figure 4-4 shows itemset-support distributions for the "Wavelets" data set, for 100, 150, and 200 citing documents, and itemset cardinalities $\chi = 2,3$. These are fairly linear for each cardinality and data set size, again indicating conformity with the Laplacian distribution. Missing values are itemset supports of zero.

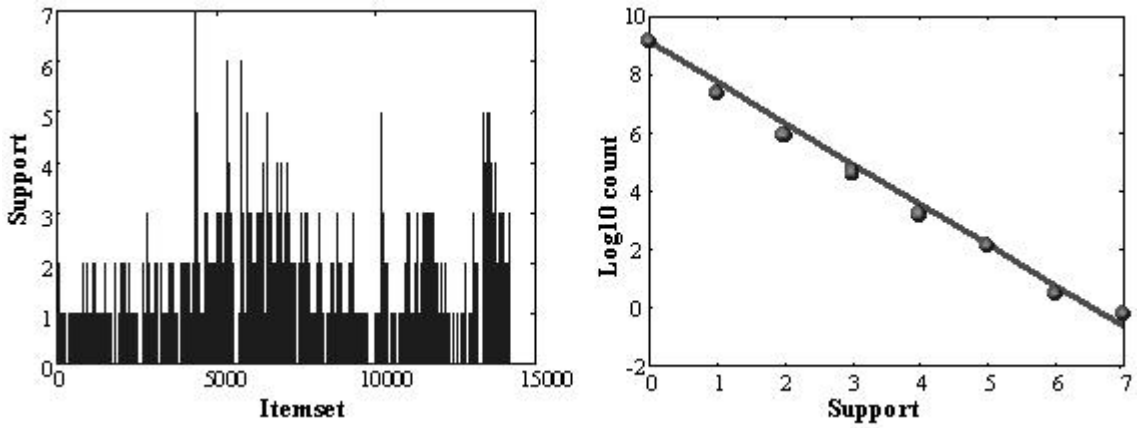


Figure 4-3: Cardinality-3 itemset-support distribution versus right-sided Laplacian for SCI "Microtubules" data set.

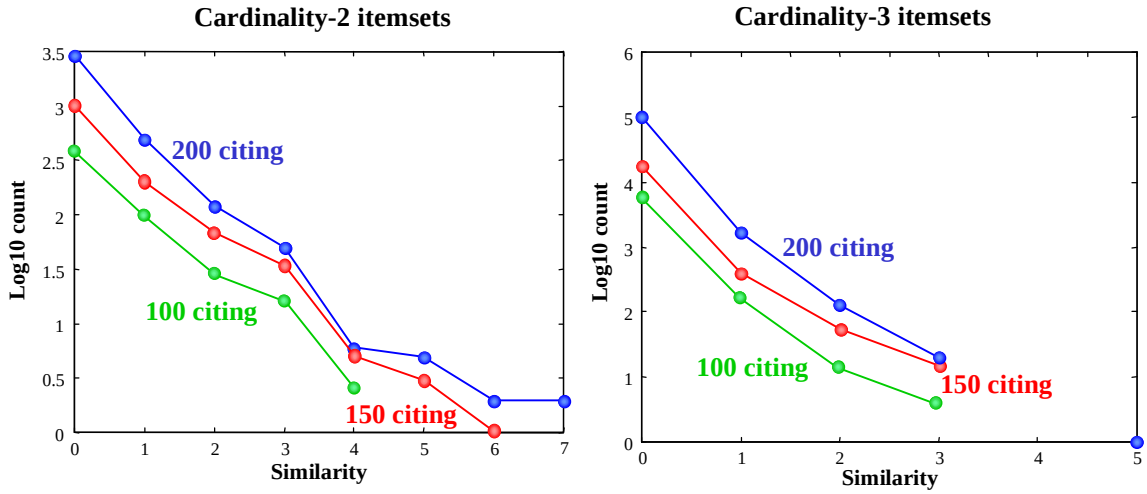


Figure 4-4: Itemset-support distributions for SCI "Wavelets" data set, cardinalities $\chi = 2,3$ with 100, 150, and 200 citing documents.

The normalization of the one-sided Laplacian distribution in Eq. (4.4) via $\lambda = \sqrt{2}/\sigma$, for standard deviation σ, enables its interpretation as a probability density. Disregarding this interpretation, a more general non-normalized distribution is

$$\tilde{p}^{+}(x) = \alpha e^{-\beta x}, \quad x \geq 0, \tag{4.5}$$

for arbitrary $\alpha, \beta > 0$. Taking the logarithm of Eq. (4.5),

$$\log\left[\tilde{p}^{+}(x)\right] = \log\alpha - \beta x. \tag{4.6}$$

Thus $\log\left[\tilde{p}^{+}(x)\right]$ is a line, with slope $-\beta$ and y-intercept $\log\alpha$.

Figures 4-5 through 4-14 show itemset supports and their distributions for the 10 SCI data sets described in Table 4-4. Results are included for both co-citations and bibliographic coupling for the data sets "Adaptive Optics," "Quantum Gravity and Strings," and "Wavelets and Brownian." Thus there are 10 total data sets.

Table 4-4: Details for SCI (Science Citation Index) data sets used in this section. Bibliographic coupling is applied in addition to co-citations for 3 of these data sets.

Data set name	Query keyword	Year(s)	Citing docs	Cited docs
Adaptive Optics	adaptive optics	2000	89	60
Collagen	collagen	1975	494	53
Genetic Algorithms and Neural Networks	genetic algorithm* and neural network*	2000	136	57
Quantum Gravity and Strings	quantum gravity AND string*	1999-2000	114	50
Wavelets (1-100)	wavelet*	1999	100	34
Wavelets (1-500)	wavelet*	1999	472	54
Wavelets and Brownian	wavelet* AND brownian	1973-2000	99	59

Figures 4-5 through 4-14 show actual support values for each itemset, along with the number of itemsets having a given support value. Each figure shows this for itemset cardinalities 2,3,4. Support distributions without missing values (for logarithms of zero) include linear least-squares fits of the data values.

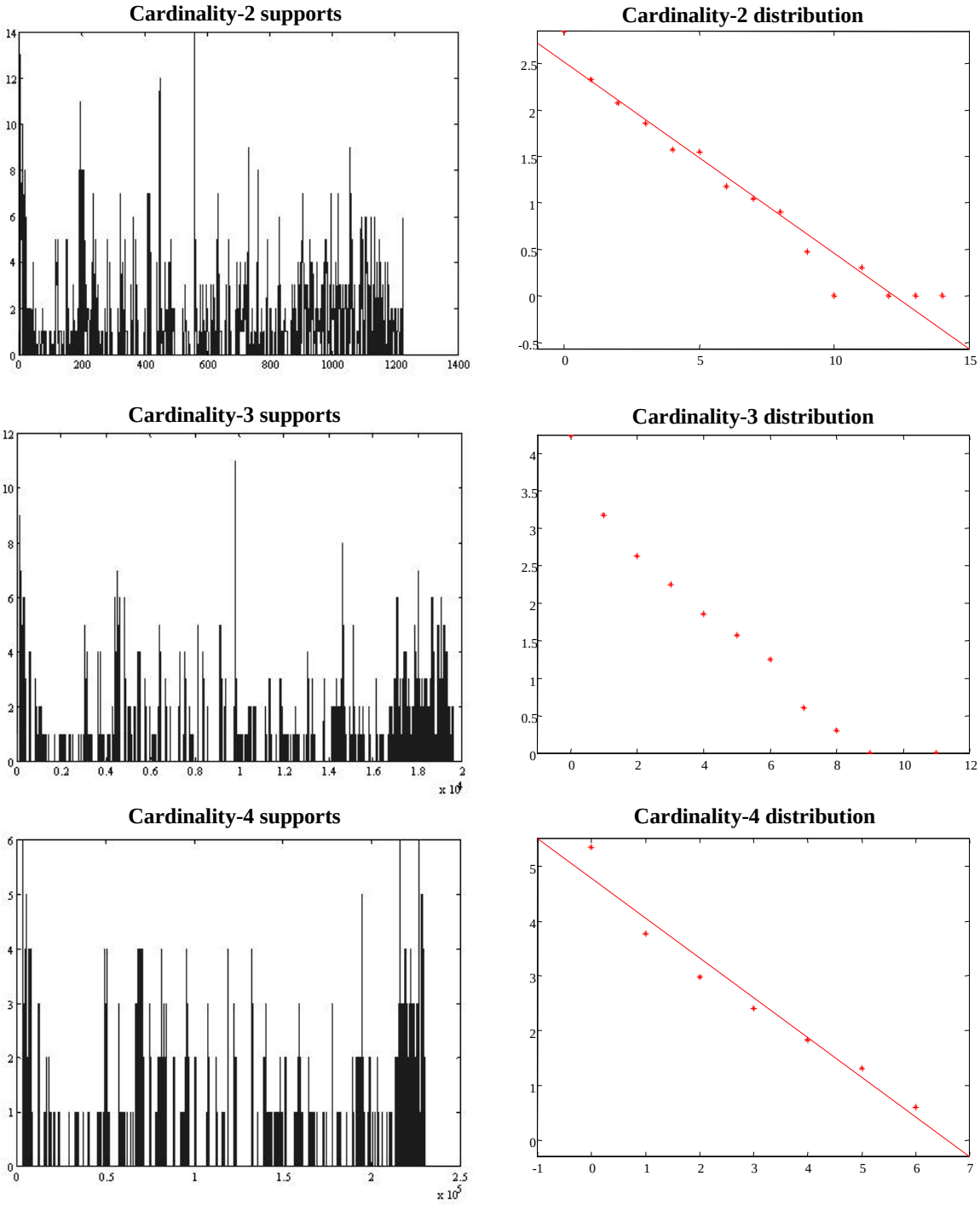


Figure 4-5: Association mining itemset supports and their distributions for cardinalities 2,3,4 for SCI "Adaptive Optics" data set.

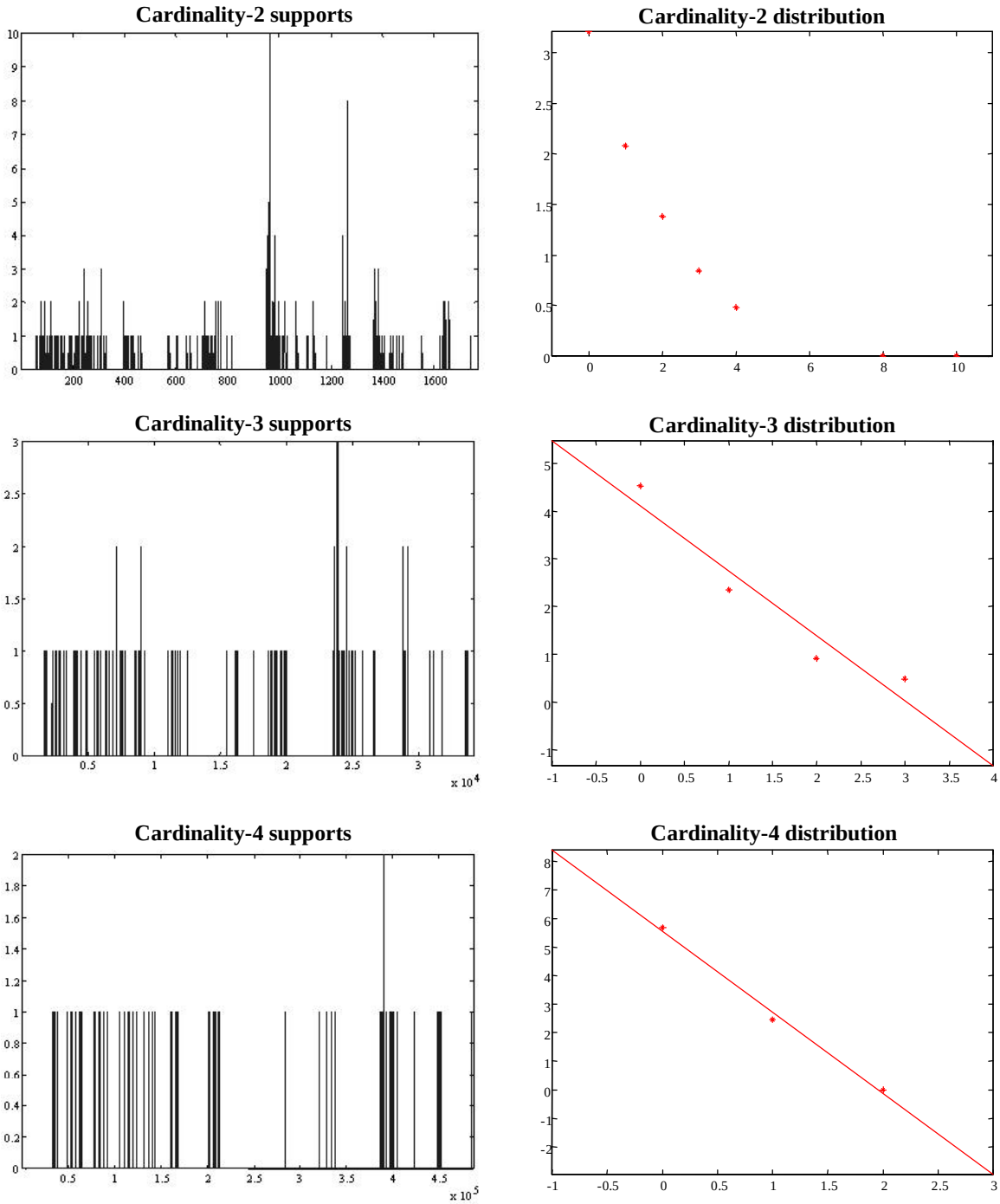


Figure 4-6: Association mining itemset supports and their distributions for cardinalities 2,3,4 for SCI "Adaptive Optics" data set (bibliographic coupling).

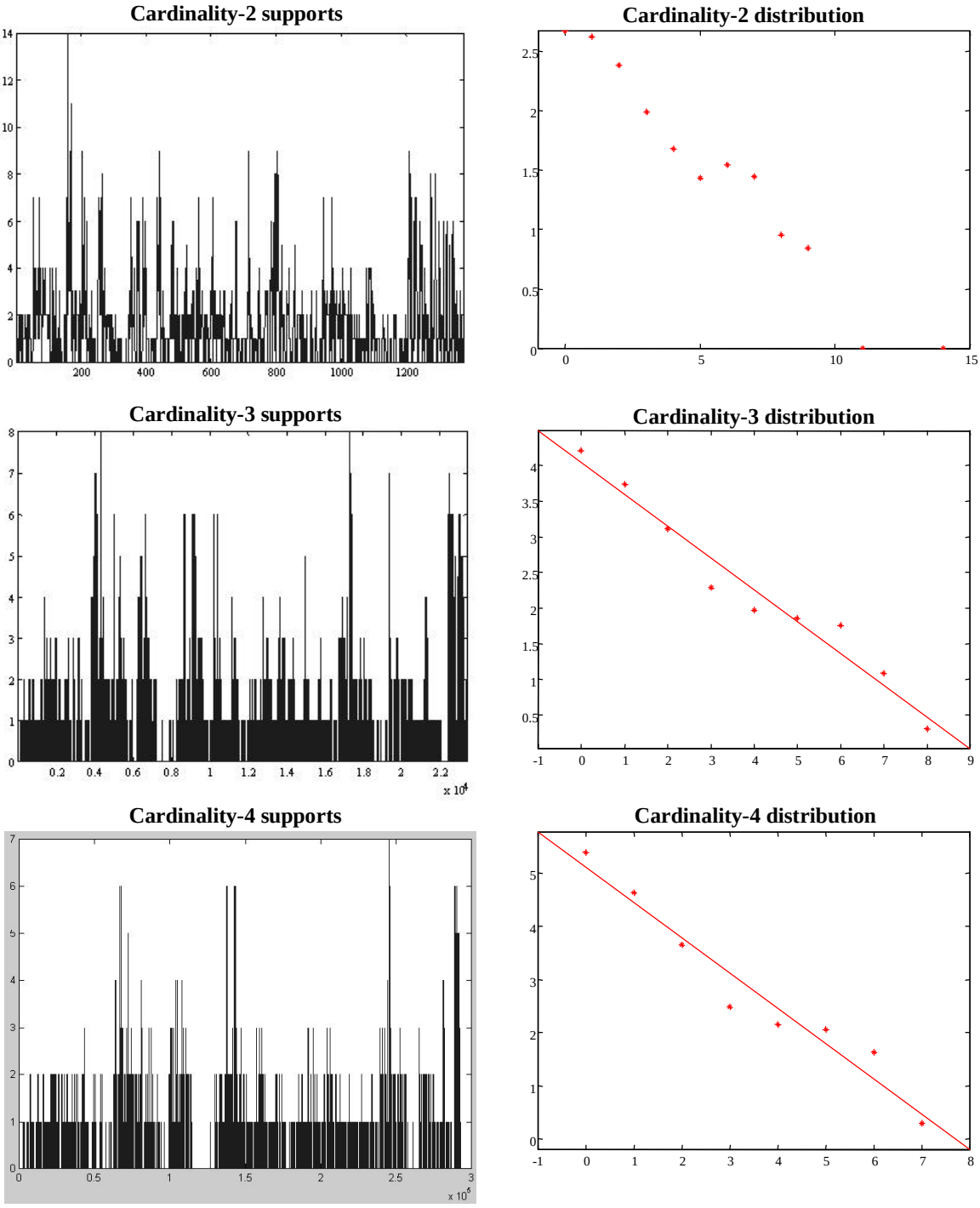


Figure 4-7: Association mining itemset supports and their distributions for cardinalities 2,3,4 for SCI "Collagen" data set.

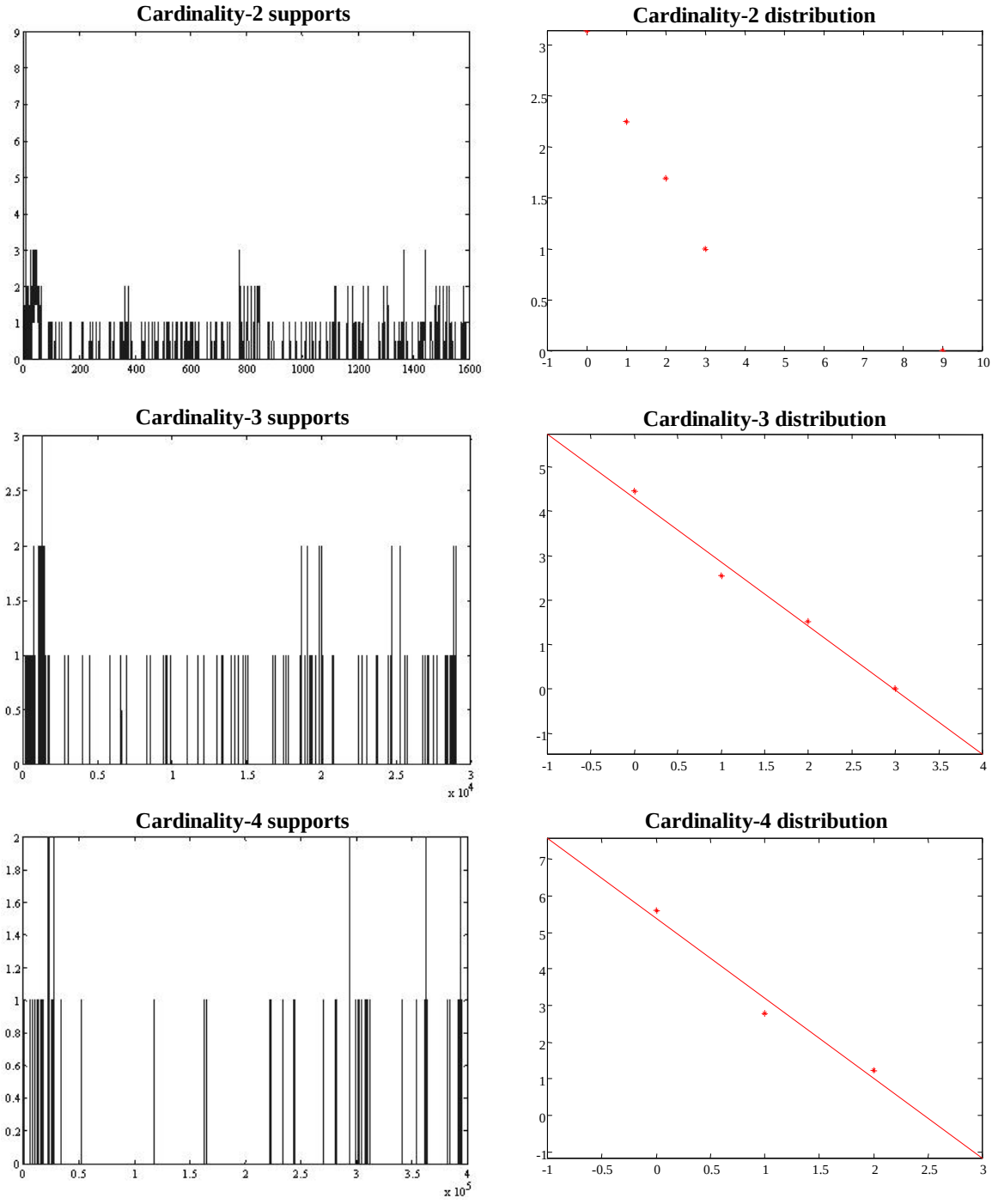


Figure 4-8: Association mining itemset supports and their distributions for cardinalities 2,3,4 for SCI "Genetic Algorithms and Neural Networks" data set.

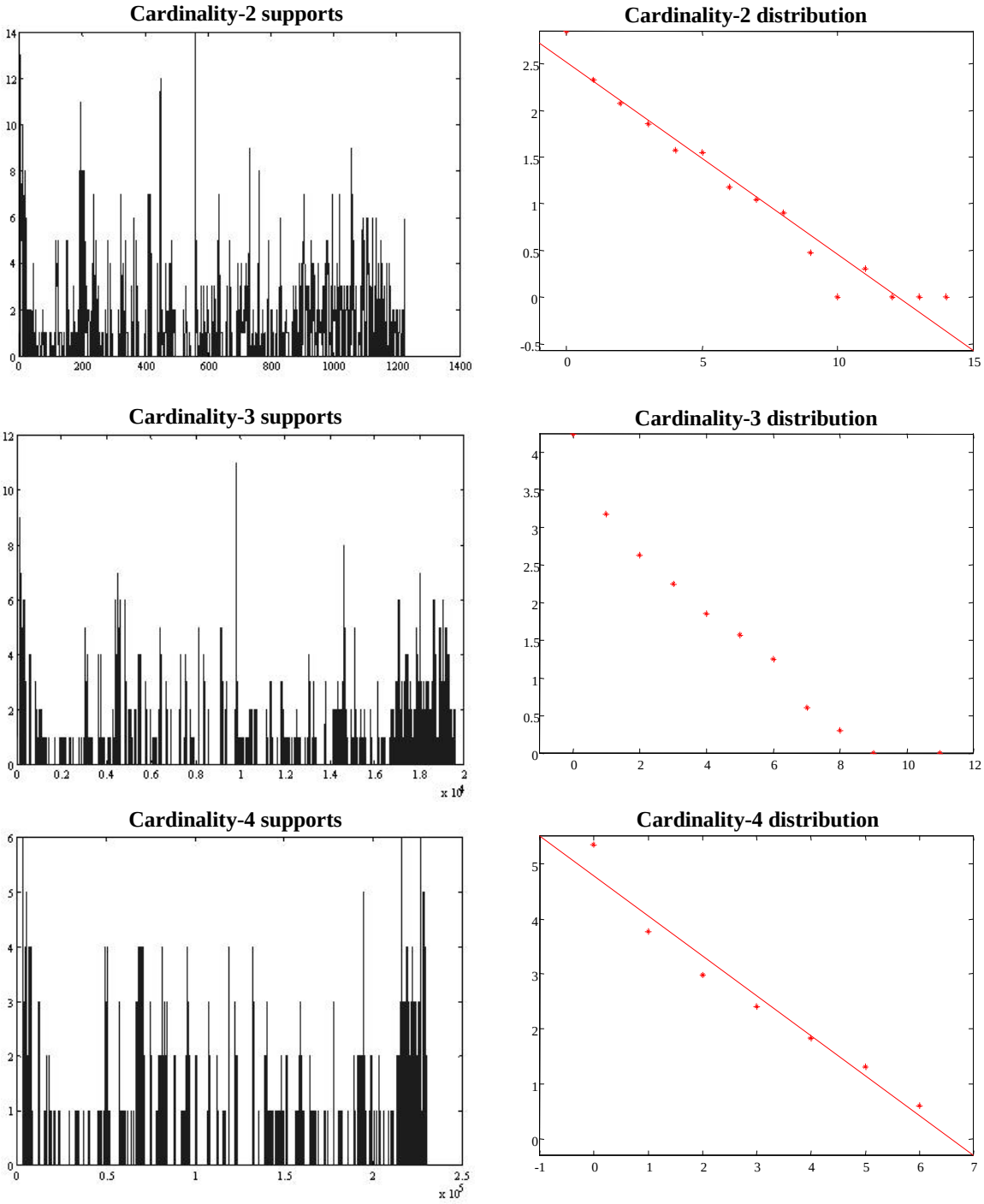


Figure 4-9: Association mining itemset supports and their distributions for cardinalities 2,3,4 for SCI "Quantum Gravity and Strings" data set.

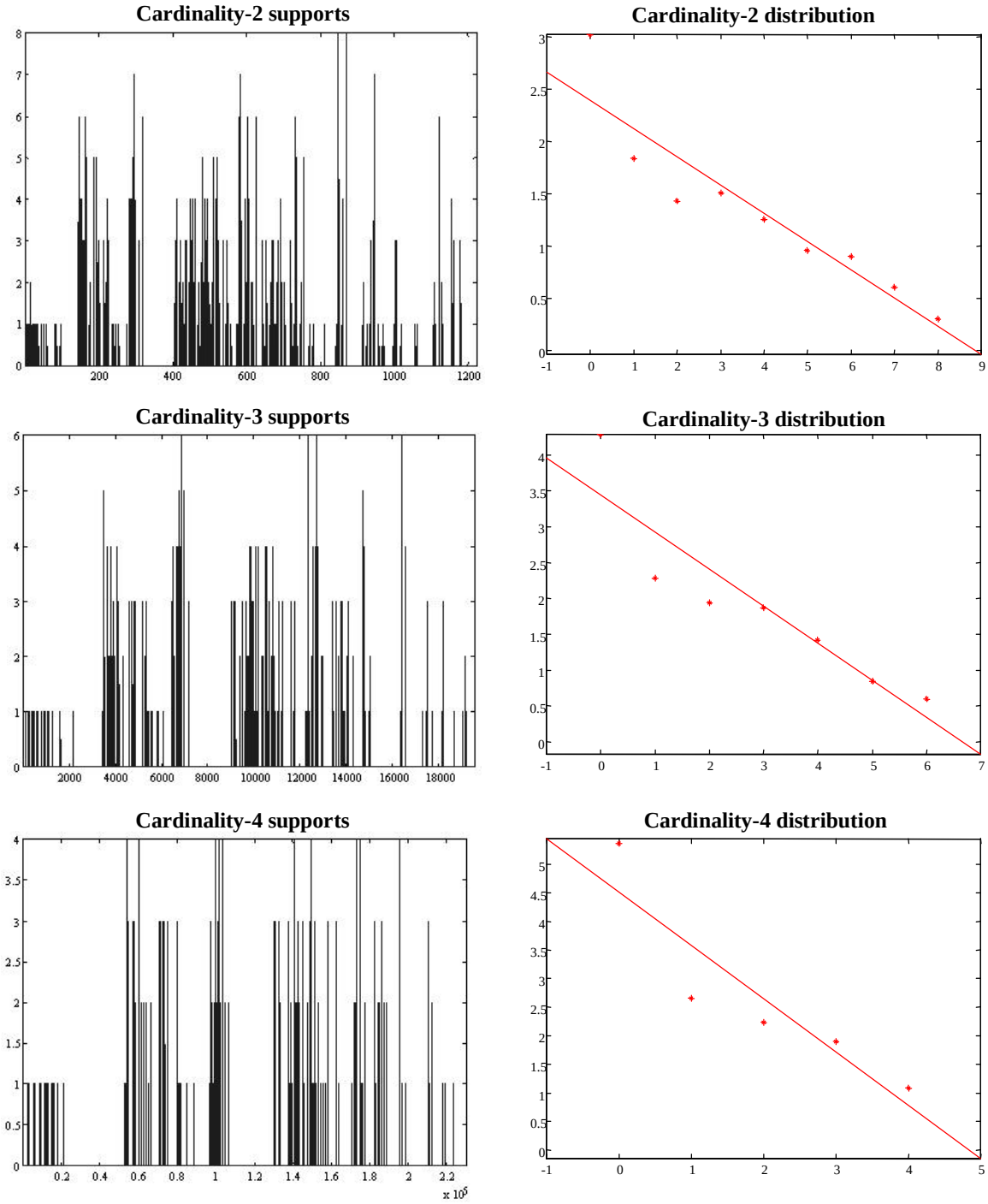


Figure 4-10: Association mining itemset supports and their distributions for cardinalities 2,3,4 for SCI "Quantum Gravity and Strings" data set (bibliographic coupling).

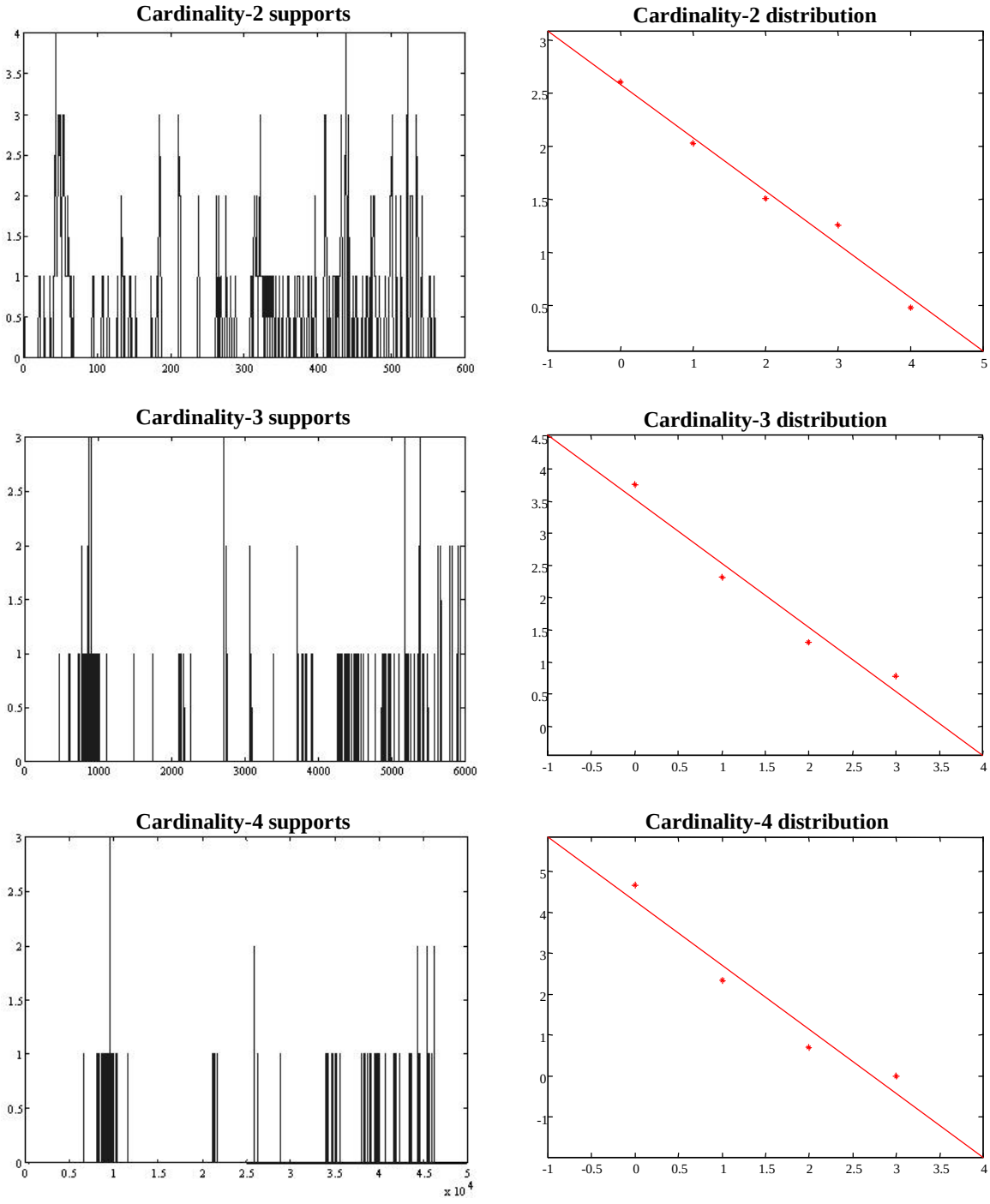


Figure 4-11: Association mining itemset supports and their distributions for cardinalities 2,3,4 for SCI "Wavelets (1-100)" data set.

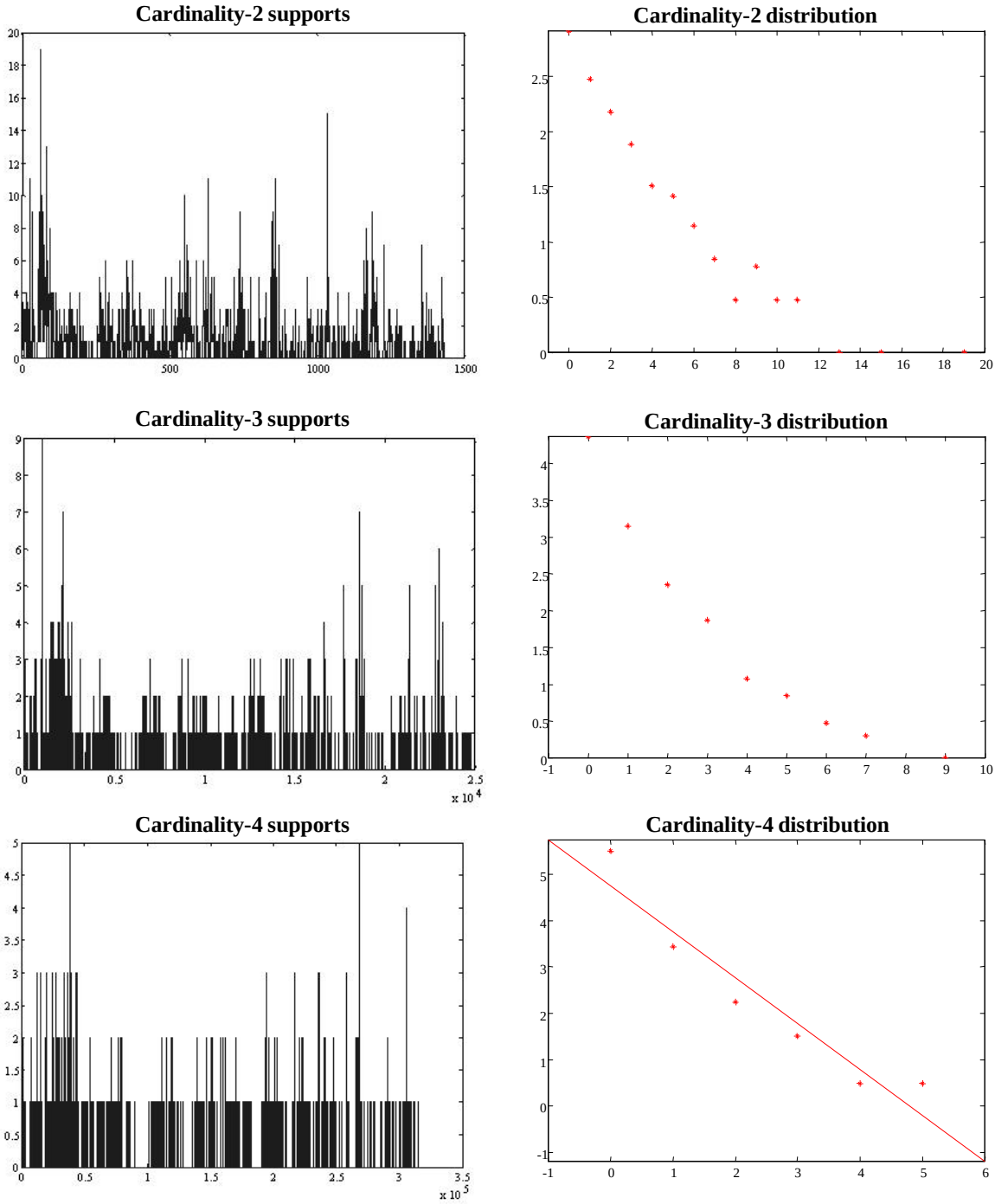


Figure 4-12: Association mining itemset supports and their distributions for cardinalities 2,3,4 for SCI "Wavelets (1-500)" data set.

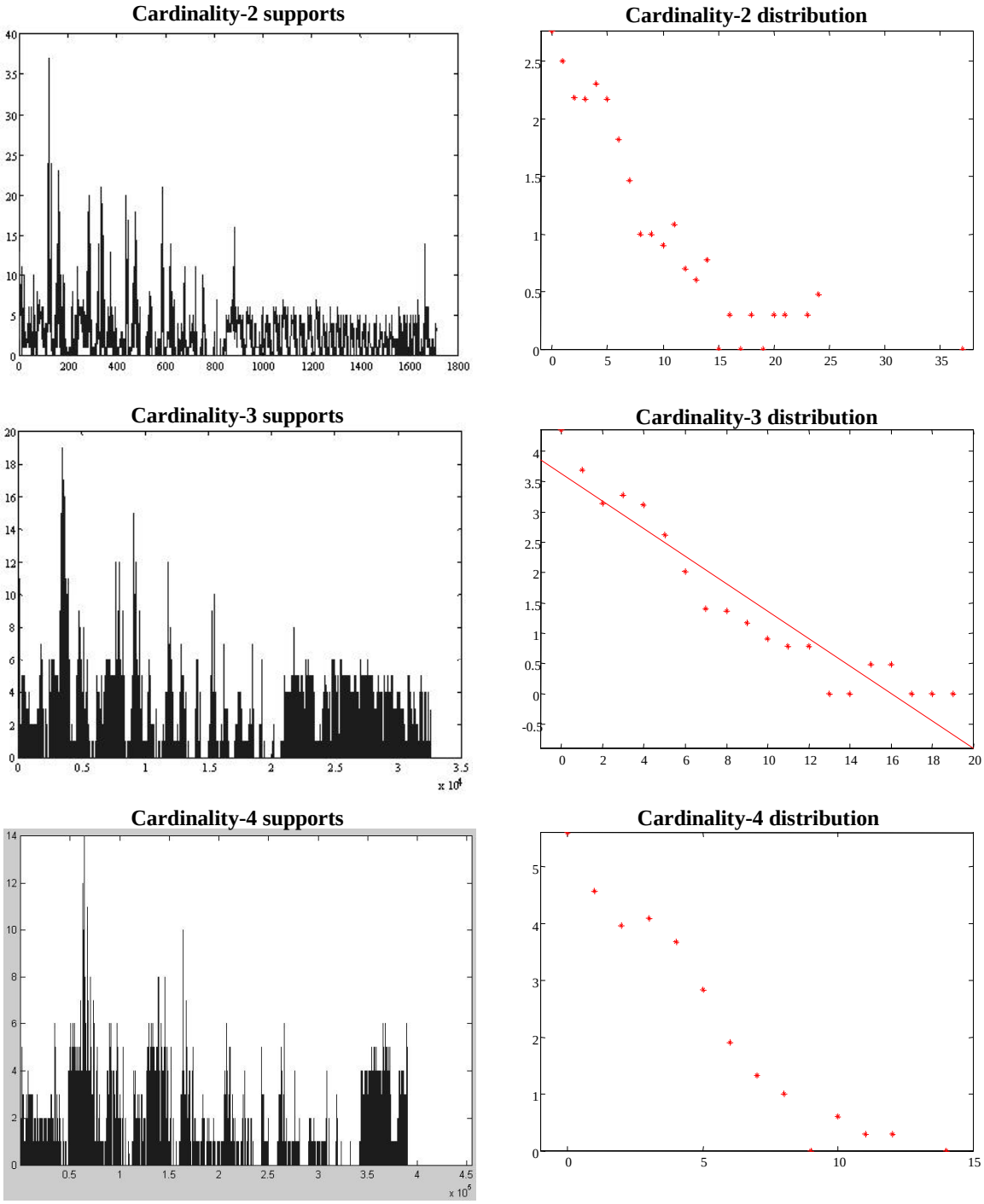


Figure 4-13: Association mining itemset supports and their distributions for cardinalities 2,3,4 for SCI "Wavelets and Brownian" data set.

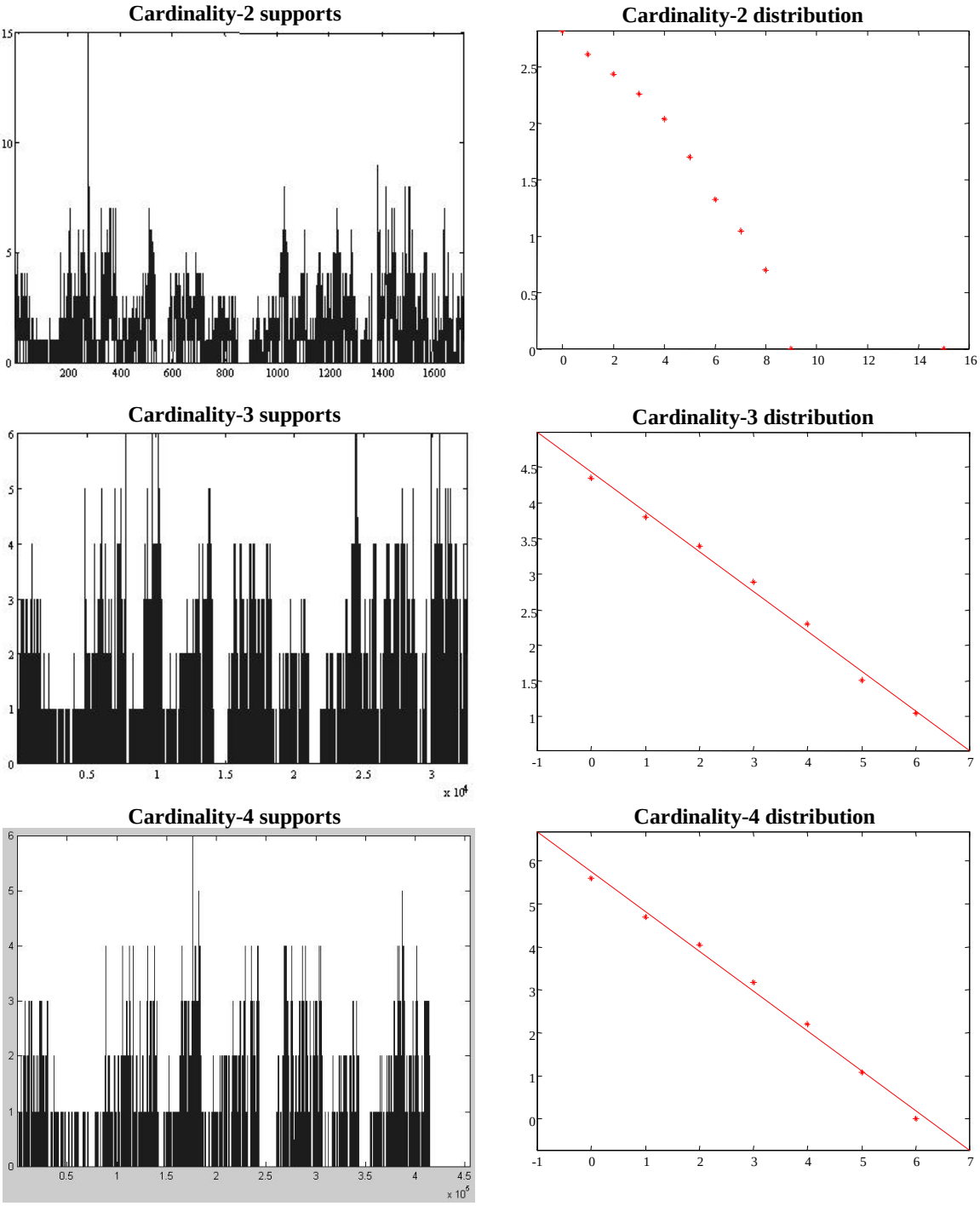


Figure 4-14: Association mining itemset supports and their distributions for cardinalities 2,3,4 for SCI "Wavelets and Brownian" data set (bibliographic coupling).

The measured itemset support distributions in Figures 4-5 through 4-14 have generally good agreement with the simple Laplacian model that I have proposed. However, the quality of the log-linear fits in the figures is data dependent. For some data sets, the fit is excellent. For others, there is significant deviation from log-linearity. For data sets with the worst fits, most log residuals are less than 0.5, corresponding to actual errors smaller than by a factor of 3. Essentially all data sets are log-linear over some range of their supports.

The exponentially decreasing nature of itemset support helps my proposed hybrid pairwise/higher-order distances produce clusters consistent with frequent itemsets. The reason is that with such a distribution, itemsets with larger supports are sparse. By the theoretical arguments I gave in Chapter 3, this makes itemsets with larger supports more likely to form exclusive clusters. That is, there are relatively few itemsets that could potentially disrupt the clustering of the most frequent ones.

The sparse nature of frequent itemsets also contributes to low computational complexity for fast frequent-itemset algorithms. These algorithms are able to rule out from further consideration supersets of the large numbers of less frequent itemsets. This supports previous experimental observations that the average time complexity of these algorithms scales linearly with problem size.

4.3 Transaction and Item Weighting

The method described in Section 4.1 of including only frequent itemsets in hybrid pairwise/higher-order distances has the potential to greatly reduce average computational complexity. However, it does not reduce the exponential worst-case complexity. This

section investigates some clever weighting of the components of pairwise distances. The new distances retain the $O(n^2)$ complexity (for n documents) enjoyed by standard pairwise distances. The hope is that the resulting clusters will be adequately consistent with frequent itemsets.

I investigate 2 weighting schemes: weighting transactions (cit*ing* documents) by their number of items (documents they cite), and weighting items (cit*ed* documents) by the number of citations they receive. The first scheme is based on the hypothesis that there is positive correlation between documents cited by large transactions and documents in frequent itemsets. That is, documents that cite many other documents tend to cite members of frequent itemsets. Thus those cited documents should be weighted more heavily, via the cardinality of the set of documents cited.

The 2nd scheme is based on a dual hypothesis. In particular, the hypothesis is that there is positive correlation between frequently cited documents and documents in frequent itemsets. Thus cited documents are weighted by their citation count, or equivalently, by the cardinality of the set of all documents citing them. Note here that itemset support is a type of *co*-citation count, versus the *citation* count used for weighting.

Consider the following simple example of transaction and item weighting. Here are the 5 example transactions (citing documents), each of unit frequency:

$$t_1 = \{1,2,3,4\}$$
$$t_2 = \{1,2,3\}$$
$$t_3 = \{1,2\}$$
$$t_4 = \{2,3,4\}$$
$$t_5 = \{4\}$$

Table 4-5 gives the citation adjacency matrix for these 5 transactions. The table also gives citation weights computed in 2 different ways: row-wise (transaction weighting) and column-wise (item weighting). For transaction weighting, each transaction (citing document) is weighted by its cardinality. That is, weight $w_i = w(t_i)$ for transaction t_i is

$$w_i = |t_i| = \sum_j a_{i,j} \,, \tag{4.7}$$

where $a_{i,j}$ is an element of the citation adjacency matrix.

Table 4-5: Example citation matrix with transaction and item weights.

Citing doc *i*	Cited doc *j*				Citing doc weight w_i
	$j = 1$	$j = 2$	$j = 3$	$j = 4$	
$I = 1$	1	1	1	1	4
$I = 2$	1	1	1	0	3
$I = 3$	1	1	0	0	2
$I = 4$	0	1	1	1	3
$I = 5$	0	0	0	1	1
	Cited doc weight v_j				
	4	4	3	3	

For item weighting, each item (cited document) is weighted by the number of citations it receives. More formally, for item *j*, the weight v_j is

$$v_j = \sum_i a_{i,j} \,. \tag{4.8}$$

When the weights w_i and v_j are normalized so that $\sum_i w_i = \sum_j v_j = 1$, they can be interpreted as probabilities in the sense of Kolmogorov.

For transaction weighting, similarities $s_{j,k}$ for each possible pair of cited documents are computed as

$$s_{j,k} = \sum_i w_i a_{i,j} a_{i,k} \,, \tag{4.9}$$

where $a_{i,j}$ and $a_{i,k}$ are distinct elements of the citation matrix. For our example, the transaction-weighted similarities for the 4 cited documents are

$$s_{1,2} = 4+3+2+0+0 = 9$$
$$s_{1,3} = 4+3+0+0+0 = 7$$
$$s_{1,4} = 4+0+0+0+0 = 4$$
$$s_{2,3} = 4+3+0+3+0 = 10$$
$$s_{2,4} = 4+0+0+3+0 = 7$$
$$s_{3,4} = 4+0+0+3+0 = 7$$

For item weighting, similarities $\tilde{s}_{j,k}$ between cited documents j and k are computed as

$$\tilde{s}_{j,k} = v_j v_k \sum_i a_{i,j} a_{i,k} \ . \tag{4.10}$$

For our example, item-weighted similarities are therefore

$$\tilde{s}_{1,2} = 4 \cdot 4 \cdot (1+1+1+0+0) = 48$$
$$\tilde{s}_{1,3} = 4 \cdot 3 \cdot (1+1+0+0+0) = 24$$
$$\tilde{s}_{1,4} = 4 \cdot 3 \cdot (1+0+0+0+0) = 12$$
$$\tilde{s}_{2,3} = 4 \cdot 3 \cdot (1+1+0+1+0) = 36$$
$$\tilde{s}_{2,4} = 4 \cdot 3 \cdot (1+0+0+1+0) = 24$$
$$\tilde{s}_{3,4} = 3 \cdot 3 \cdot (1+0+0+1+0) = 18$$

As a comparison, standard pairwise similarities $\hat{s}_{j,k}$, in which $w_i = v_j = 1$, are computed as

$$\hat{s}_{j,k} = \sum_i a_{i,j} a_{i,k} \ . \tag{4.11}$$

For our example, standard pairwise similarities are then

$$\hat{s}_{1,2} = 3$$
$$\hat{s}_{1,3} = 2$$
$$\hat{s}_{1,4} = 1$$

$$\hat{s}_{2,3} = 3$$
$$\hat{s}_{2,4} = 2$$
$$\hat{s}_{3,4} = 2$$

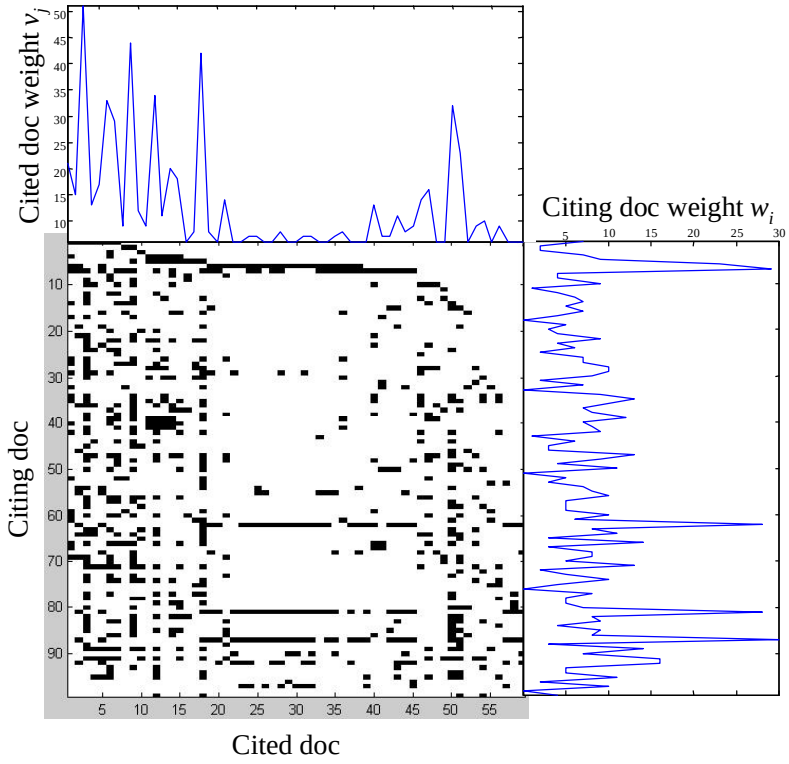
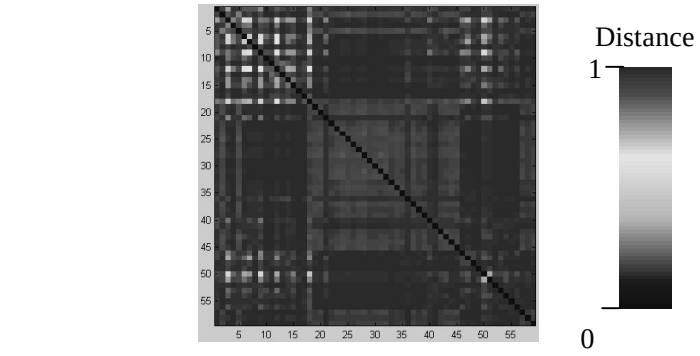
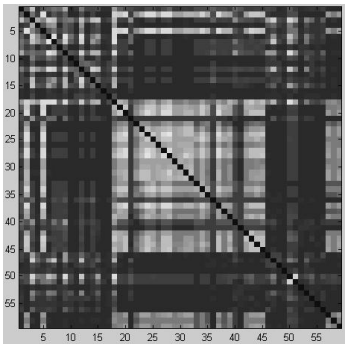


Figure 4-15: Citation matrix with transaction and item weights for "Wavelets and Brownian" data set.

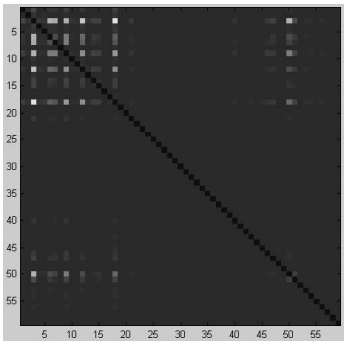
Figure 4-15 shows the citation adjacency matrix for the SCI "Wavelets and Brownian" data set. For this data set, the query keyword is "wavelet* AND brownian" and the years are 1972-2000. The first 99 documents matching the query cite a total of 1892 unique documents. I filter documents cited less than 6 times, yielding the adjacency matrix columns for the remaining 59 highly cited documents. Along with the adjacency matrix, Figure 4-15 also shows corresponding weights for transactions (citing documents) and items (cited documents).



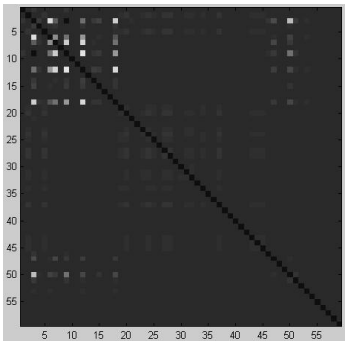
(a) Pairwise



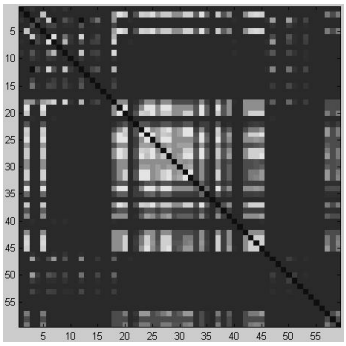
(b) Transaction weighting



(c) Item weighting



(d) (Cardinality 3)^4



(e) (Cardinality 4)^4

Figure 4-16: Distance matrices for "Wavelets and Brownian" data set: (a) standard pairwise, (b) transaction weighting, (c) item weighting, (d) hybrid from cardinality-3 itemsets, and (e) hybrid from cardinality-4 itemsets.

The resulting distance matrices are shown in Figure 4-16. In this figure, part (b) is from Eq. (4.9) transaction weighting, and part (c) is from Eq. (4.10) item weighting. Both of these are significantly different from the Eq. (4.11) standard pairwise distances, shown in part (a).

Figure 4-16 compares transaction and item weighting to distances computed from cardinality-3 and cardinality-4 itemset supports. The cardinality-3 similarities for part (d) are computed as

$$\text{(Cardinality 3)\^{}4} \equiv s_{i,j} = \sum_{\substack{i,j \in I, \\ |I|=3}} [\zeta(I)]^4, \tag{4.12}$$

while the cardinality-4 similarities shown in part (e) are computed as

$$\text{(Cardinality 4)\^{}4} \equiv s_{i,j} = \sum_{\substack{i,j \in I, \\ |I|=4}} [\zeta(I)]^4. \tag{4.13}$$

The distance matrices for transaction weighting and cardinality-4 itemsets are vaguely similar, as are the matrices for item weighting and cardinality-3 itemsets. Each of these 4 matrices also have elements in common with the one for standard pairwise distances.

However, these 5 types of distance computations lead to significantly different clusterings. We see this in Figures 4-17 through 4-21, which show complete-linkage clustering dendrograms for the 5 types of distances. The dendrograms include the single most frequent cardinality-3 and cardinality-4 itemsets, which are {3, 7, 9}▲ and {3, 7, 9, 12}●, respectively.

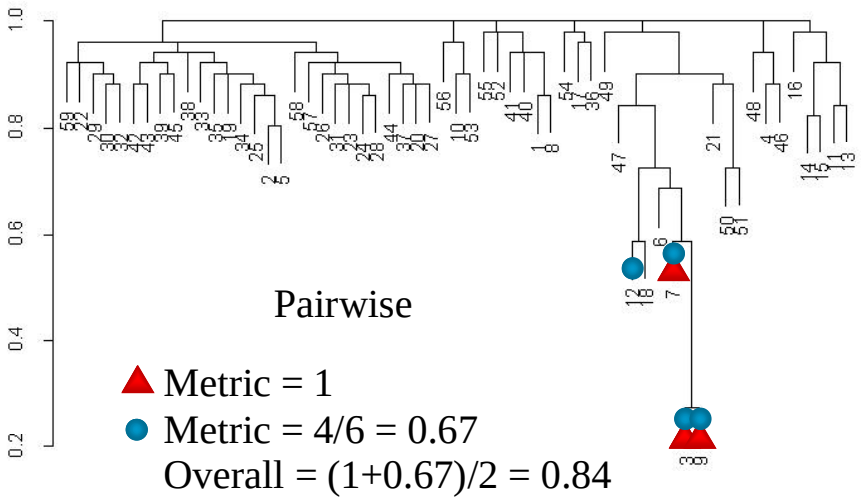


Figure 4-17: Standard pairwise similarities versus frequent itemsets for complete-linkage clustering of "Wavelets and Brownian" data set.

The clustering for cardinality-3 similarities (Figure 4-20) is the most consistent with the frequent itemsets, which is reflected by its overall clustering metric value of unity. The overall metric value for transaction weighting (Figure 4-18) is slightly better than for cardinality-4 similarities (Figure 4-21). However, in a sense, clustering for cardinality-4 similarities is more consistent with the frequent itemsets, since it has a cluster that contains only items from the 2 itemsets, unlike any cluster for transaction weighting. While transaction weighting is a slight improvement over standard pairwise (Figure 4-17), the metric value for item weighting (Figure 4-19) is significantly smaller than for standard pairwise.

Tables A3-1 through A3-5 in Appendix C show clustering metric results for the transaction and itemset weighting schemes for reducing computational complexity. These correspond to Eqs. (4.9) and (4.10), respectively. The clustering metrics are computed for the SCI data sets described in Table 4-6. For the data set "Wavelets and

Brownian," results are included for both co-citations and bibliographic coupling, yielding a total of 5 data sets.

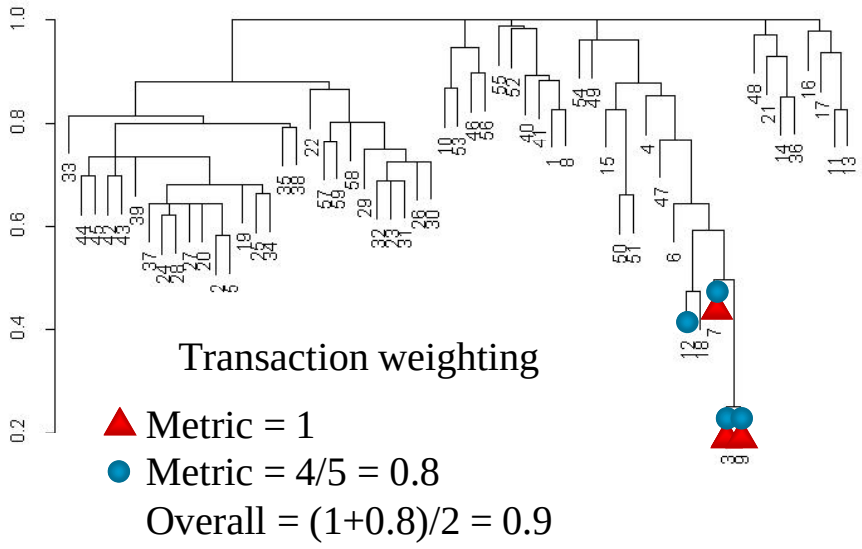


Figure 4-18: Transaction weighting similarities, itemset-matching metric for complete-linkage clustering of "Wavelets and Brownian" data set.

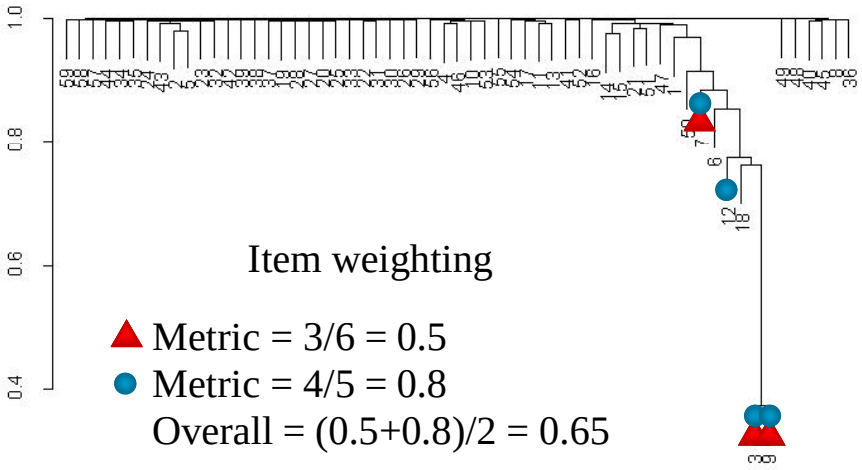


Figure 4-19: Item weighting similarities, itemset-matching metric for complete-linkage clustering of "Wavelets and Brownian" data set.

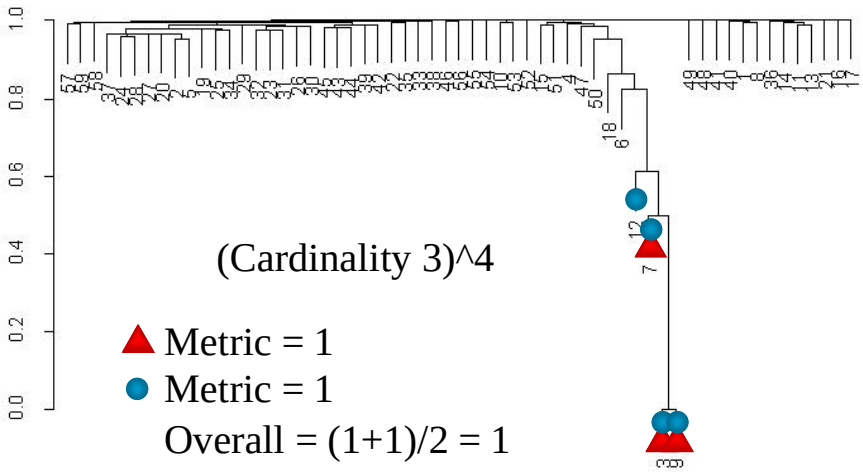


Figure 4-20: Cardinality-3 similarities, itemset-matching metric for complete-linkage clustering of "Wavelets and Brownian" data set.

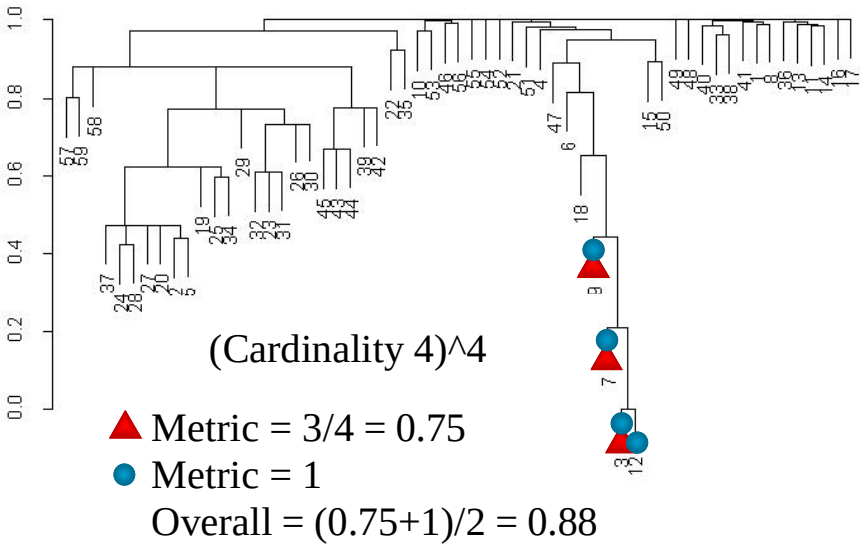


Figure 4-21: Cardinality-4 similarities, itemset-matching metric for complete-linkage clustering of "Wavelets and Brownian" data set.

Table 4-6: Details for SCI data sets used in this section. Bibliographic coupling is applied in addition to co-citations for one of these data sets.

Data set name	Query keyword	Year(s)	Citing docs	Cited docs
Collagen	collagen	1975	494	53
Quantum Gravity and Strings	quantum gravity AND string*	1999-2000	114	50
Wavelets (1-500)	wavelet*	1999	472	54
Wavelets and Brownian	wavelet* AND brownian	1973-2000	99	59

The clustering metric results in Tables A3-1 through A3-5 are summarized in Table 4-7 (for transaction weighting) and Table 4-8 (for item weighting). Transaction weighting has a higher metric value about twice as often as standard pairwise distances. While this is somewhat encouraging, it represents considerably lower consistency with frequent itemsets in comparison to hybrid pairwise/higher-order distances, which have higher metric values nearly 7 times more often than standard pairwise distances.

Table 4-7: Clustering metric comparisons for transaction weighting (T.W.) versus standard pairwise (P.W.) distances.

Data set	T.W.=P.W.	T.W.>P.W.	T.W.<P.W.	Cases
1	0	17	1	18
2	3	7	8	18
3	7	4	7	18
4	3	10	5	18
5	1	17	0	18
Totals	14	55	21	90

Performance for item weighting is quite poor in terms of the itemset-matching clustering metric, as shown in Table 4-8. In particular, the metric value for standard

pairwise distances is larger than for item weighting distances about one and a half times as often.

Table 4-8: Clustering metric comparisons for item weighting (I.W.) versus standard pairwise (P.W.) distances.

Data set	I.W.=P.W.	I.W.>P.W.	I.W.<P.W.	Cases
1	0	2	16	18
2	3	7	8	18
3	6	0	12	18
4	1	5	12	18
5	3	12	3	18
Totals	13	26	51	90

In interpreting the relatively poor itemset-matching performance of transaction and item weighting, recall that itemset support (higher-order co-citation count) is a joint property among a set of items. It appears that pairwise co-citations contain insufficient information for approximating larger-cardinality itemsets (higher-order co-citations). There may be no way of weighting combinations of mere pairwise co-citations to adequately approximate higher-order co-citations.

This section concludes Chapter 4. In this chapter, I have investigated methods for reducing complexity for distance computations. I demonstrated that my new hybrid pairwise/higher-order distances are consistent with fast algorithms for computing frequent itemsets. In particular, the exclusion of less frequent itemsets has a very small effect on cluster itemset matching. Thus the hybrid distances are computationally tractible as well as convenient for user interaction and analysis.

This chapter also showed for the first time that citation itemset supports generally follow a Laplacian or decreasing exponential distribution. This is consistent with the empirical linear scaling with problem size that has been previously been reported for computing frequent itemsets. The sparseness of more frequent itemsets for this distribution also contributes to the consistency of clusters and frequent itemsets for hybrid pairwise/higher-order distances.

Furthermore, this chapter provided empirical evidence that weighting of the components of mere pairwise distances is insufficient for clustering frequent itemsets. This suggests that the higher-order citations employed in my new hybrid distances are necessary.

The next chapter applies the new hybrid pairwise/higher-order distances to visualizations of the minimum spanning tree.

Chapter 5

Minimum Spanning Tree with Higher-Order Co-Citations

The minimum spanning tree has previously been proposed for visualizing document collections, with the distances between documents computed from co-citations [Chen99a][Chen99b]. The tree shows the minimal set of essential links among documents, which is interpreted as the network of direct influences within the collection. This provides useful analysis for information retrieval. Branches in the tree correspond to bifurcations of ideas in the evolution of science, with highly influential documents appearing near the center of the network, and the emerging research front represented as documents on the fringes.

This chapter applies the hybrid pairwise/higher-order distances proposed in Chapter 3 to minimum spanning tree visualizations. The new distance methodology reduces distances among members of association mining frequent itemsets (higher-order co-citations). This results in more direct influences among the frequent itemset members in the minimum spanning tree visualization, and generally increases their influence within the entire tree.

I propose 3 new metrics for evaluating minimum spanning trees with respect to various distance formulas. These all measure changes in network influence for documents in frequent itemsets. The first metric measures the extent to which frequent-itemset documents directly influence each other, through the number of connected

components they form in the tree. The 2nd metric measures the direct influence of individual members of frequent itemsets, through their graph-theoretic degree. The 3rd metric is as analysis of the influence of frequent itemset members within the entire network.

I apply each of the 3 new metrics to data sets extracted from the SCI (Science Citation Index). In particular, the metrics help test the effects of hybrid pairwise/higher-order distances on the minimum spanning tree.

I also introduce a new minimum spanning tree visualization that employs the 3rd spatial dimension. It applies the wavelet transform in estimating multiple-resolution densities of the minimum spanning tree vertices. A density is visualized as a height surface, and the tree is then embedded in this surface. The density surface is analogous to a landscape, in which high concentrations of documents appear as peaks and low concentrations as valleys. The ability of humans to readily interpret landscapes allows rapid identification of peaks at various resolutions, as hierarchical document clusters.

The next section describes the minimum spanning tree problem, along with algorithms for computing them. Section 5.2 then describes an algorithm for placing minimum spanning tree vertices in the plane, so that the tree can be visualized. In Section 5.3, I propose itemset-based metrics for minimum spanning trees. Section 5.4 then applies these metrics to various SCI data sets. Finally, Section 5.5 introduces the new wavelet-based minimum spanning tree density landscape visualization.

5.1 Minimum Spanning Tree

A minimum spanning tree for an undirected graph is the minimum-distance tree that connects all vertices in the graph. Thus vertices are connected to one another by the smallest possible distance between them. In terms of hypertext systems with distances based on co-citation, a minimum spanning tree edge is interpreted as the most direct influence between 2 documents.

Stated more formally, given a weighted graph $G = (V, E)$ with vertices V, edges E, and weight $w(u,v)$ of the edge $(u,v) \in E$, the minimum spanning tree is a set of edges $T \subseteq E$ that connects all vertices V such that the sum of the edge weights

$$w(T) = \sum_{(u,v) \in T} w(u,v) \tag{5.1}$$

is minimized. The minimum spanning tree need not be unique in the case of identical weight values for graph edges $(u,v) \in E$. The set of edges T is acyclic – if there were a cycle, some edge in the cycle could be removed, thereby reducing the total weight while maintaining total connectivity.

The most well known algorithms for computing the minimum spanning tree are Kruskal's algorithm and Prim's algorithm. Both can be easily implemented in $O(E \log V)$ time [Corm96]. In fact, Prim's algorithm can be made to run in $O(E + V \log V)$ time. This is an improvement for our completely connected distance graphs, in which $E = \Theta(V^2)$, so that Prim's algorithm becomes $O(V^2)$.

Minimum spanning tree algorithms generally follow a greedy approach, which in this case is guaranteed to be optimal. These greedy algorithms grow the tree one edge at a time, while maintaining a subset of the tree. Each step of the algorithm determines an

edge that is *safe*, i.e. one that is guaranteed to be a subset of the minimum spanning tree. There is a theorem in graph theory that gives the condition for safe edges. It is defined in terms of a *cut* of the graph, which is illustrated in Figure 5-1.

Figure 5-1 shows a cut $(S, V-S)$ in a graph $G=(V,E)$. The cut is a partition that separates vertices V into the sets S and $V-S$. An edge is said to *cross* the cut $(S, V-S)$ if it is incident on both an element of S and an element of $V-S$. Further, the cut $(S, V-S)$ *respects* the set of edges A if it crosses no edge in A. A *light edge* is an edge of minimum weight that crosses the cut. The theorem then guarantees the safety of an edge (u,v) in the following manner: Given the connected graph $G=(V,E)$, a subset A of the minimum spanning tree, a cut $(S, V-S)$ that respects A, and a light edge (u,v) crossing $(S, V-S)$, then (u,v) is a safe edge.

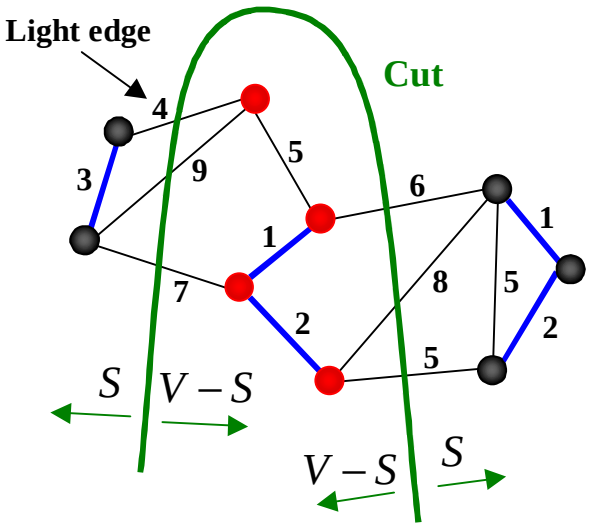


Figure 5-1: Cut $(S, V-S)$ of graph for computing the minimum spanning tree. The cut partitions graph vertices V into sets S and $V-S$, while respecting the set of edges A, shown in blue.

Kruskal's algorithm and Prim's algorithm differ in the rule in which they determine safe edges. Kruskal's algorithm maintains a set of edges A that forms a forest. A safe edge is the minimum-weight edge that connects 2 trees in the forest. In Prim's algorithm, the set A forms a single tree rather than a forest. A safe edge added to A is then the minimum-weight edge that connects the tree to a vertex not already in it.

This section introduced the minimum spanning tree problem, and described algorithms for computing minimum spanning trees. The next section describes an algorithm for computing spatial coordinates for minimum spanning tree vertices, for the purpose of visualization.

5.2 Minimum Spanning Tree Vertex Placement

While the computation of the minimum spanning tree is straightforward, it remains to put the tree in a form appropriate for visualization. The resulting visualized tree has the interpretation as a network of influences among the hyperlinked documents. Spatial coordinates must be induced for the minimum spanning tree vertices in order to apply computer graphics. That is, we must spatialize this inherently non-spatial object. The only information at our disposal is the topology of the minimum spanning tree graph, and the corresponding edge weights.

A classical method for visualizing distance data is multidimensional scaling [Vena94]. The distances are often generated from high-dimensional data, so that multidimensional scaling is seen as a method of dimensionality reduction. That is, it is a method of projecting high dimensional data into lower dimensions, while trying to

preserve distances among the data items. In the case of our co-citations, document distances are computed from of high-dimensional inner products of the citation adjacency matrix columns (generalized inner products for higher-order co-citations).

Multidimensional scaling is seen as a nonlinear competitor to principal component analysis. The classical form of multidimensional scaling with Euclidean distances is equivalent to the first k principal components, where k is the dimensionality of the visualization. Like principal component analysis, multidimensional scaling attempts to compute linear combinations of the original dimensions that maximize data variance. Thus it blends together those dimensions that seem to be well correlated. Standard algorithms for multidimensional scaling are slow, e.g. $O(n^3)$ or $O(n^4)$. Also, most algorithms are non-iterative, that is, they must be started from the beginning each time, which can be a disadvantage for many applications.

Multidimensional scaling attempts to map data objects $i = 1,2,\ldots,n$ to spatial points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ (e.g. in the plane) such that the original distances are approximately maintained. A *stress function* measures the fitness of the approximation, i.e. the fitness between the original distances and the distances for the mapped points. The simplest stress function $f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ is the sum of the squares of the residual, or

$$f(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n) = \sqrt{\sum_{j,k} \left(d_{j,k} - \left\| \mathbf{x}_j - \mathbf{x}_k \right\| \right)^2}, \tag{5.2}$$

where $\|\bullet\|$ is the vector norm, usually Euclidean. Minimization of the stress function usually employs some form of gradient descent, though simulated annealing is sometimes applied to avoid getting stuck in local minima.

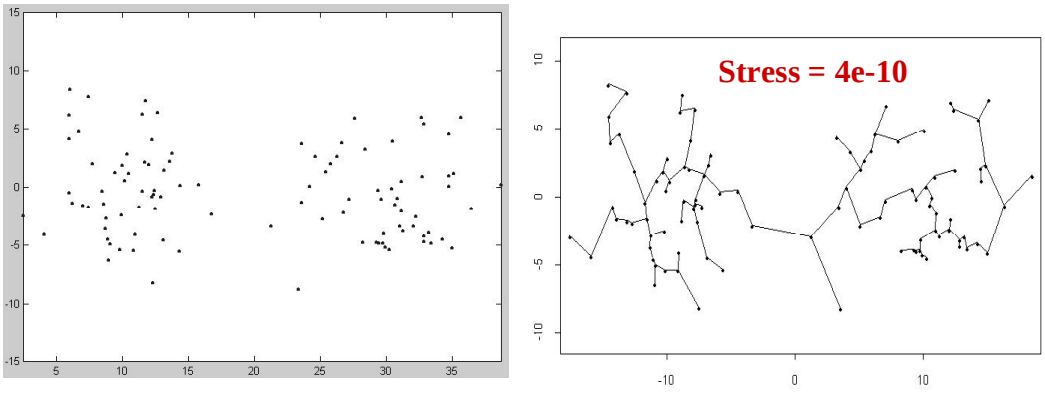


Figure 5-2: Multidimensional scaling for synthetic distances with Euclidean norm.

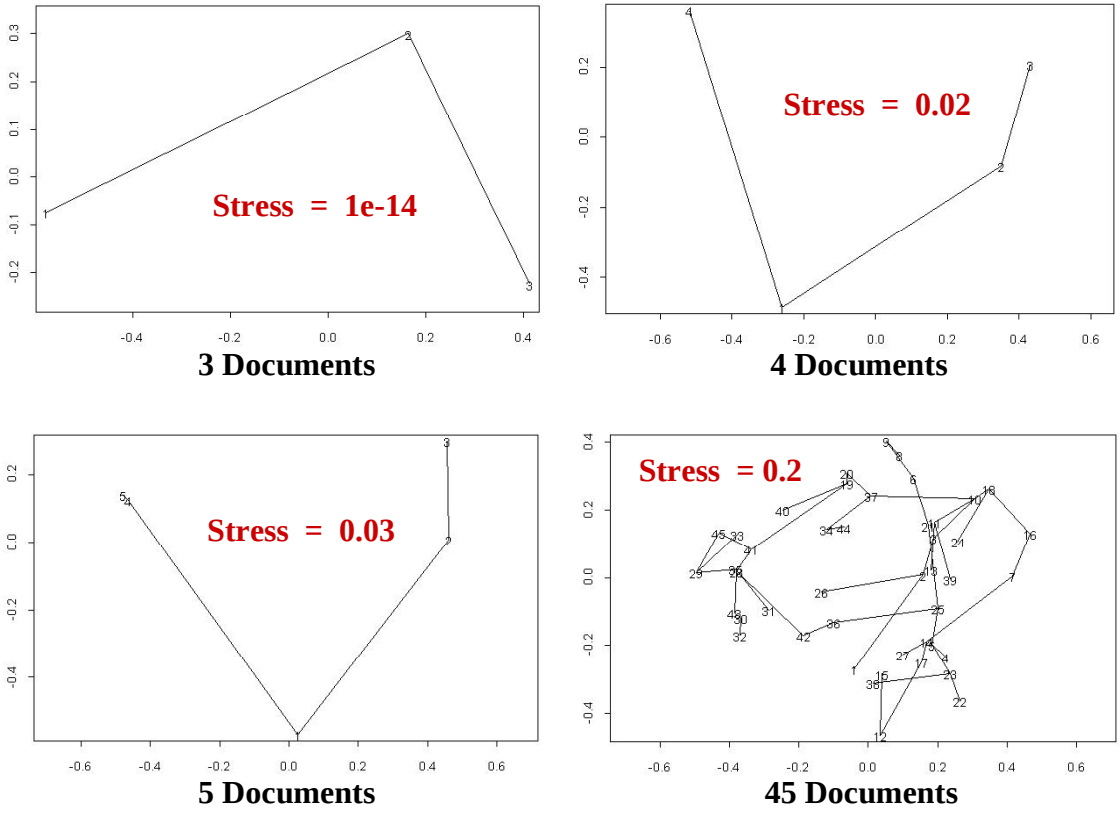


Figure 5-3: Multidimensional scaling fails for real-life SCI documents.

A serious problem occurs for multidimensional scaling when the original distances are not Euclidean, since there is no way to preserve the distances in the projection to lower dimensions. This is shown in Figures 5-2 and 5-3, which include both the projected points and the minimum spanning tree. Figure 5-2 is multidimensional scaling for synthetically generated points whose distances are Euclidean. The resulting stress is very low, indicating a close match to the original distances, and the spatial layout of the projected points is excellent.

Figure 5-3 shows multidimensional scaling for the SCI (Science Citation Index) "Microtubules" data set. Stress for only 3 documents is very low, since 3 points whose distances obey the triangle inequality can always be drawn in the plane. But stress increases dramatically with the addition of a 4^{th} point, and continues to increase as points are added. The minimum spanning tree visualization for 45 documents is quite poor, containing many confusing crossed edges.

Multidimensional scaling is one of many optimization-based algorithms for dimensionality reduction or graph node placement. Given an error metric for node placement, these attempt to place nodes so as to minimize the metric. A frequently applied class of algorithms models a system of forces and mechanical springs in placing nodes, and was developed initially for VLSI and popularized by the work of Eades [Eade84].

In so-called spring or force-displacement models, springs corresponding to graph edges generate forces that move vertices to better match the springs' characteristic lengths. This in turn reduces the spring-length dependent stress, interpreted as total system energy. A direct implementation of such a model is a form of the well-known *N*-

body simulation from celestial and atomic physics. It has time complexity $\Theta(N^2)$, since each object interacts with every other one.

But classical spring models still attempt to minimize distance errors among all pairs of items, which performs poorly for significant deviations from Euclidean distances. I employ a type of spring model that abandons the idea of minimizing distance error over all pairs [Fruc91]. That is, it makes no explicit attempt to minimize a stress formula.

Designed for general undirected graphs and not just fully connected ones, it models attractive spring forces only for edges actually in the graph. These forces cause adjacent vertices to be drawn towards on another. There are repulsive forces among all vertices, which prevent them from collapsing onto one another, and distribute them in the plane. The heuristic has been shown to generate graph layouts that follow generally accepted aesthetic criteria, such as minimizing edge crossings, distributing vertices evenly, and reflecting underlying symmetries. Straightforward implementation of the heuristic has time complexity $\Theta(V^2 + E)$ for each iteration, but there is an improvement that allows it to run in time $\Theta(V + E)$ per iteration.

The heuristic begins with an initial set of plane coordinates for the vertices (documents), usually random. The algorithm then iteratively applies 3 steps: computing the effect on vertices by repulsive forces, computing the effect on vertices by attractive forces, and updating the temperature value. In computing forces, consider the ideal distance κ between vertices, which is

$$\kappa = C\sqrt{\frac{A_{\text{frame}}}{n}}\,. \tag{5.3}$$

Here A_{frame} is the area of the frame that is to contain the visualized graph, and C is an experimentally determined parameter for how the vertices fill the frame. In my experiments, I found that an explicit frame can be abandoned altogether, making C unnecessary and the value of κ arbitrary.

The magnitude of the repulsive force $f_r(d)$ between 2 vertices separated by distance d is then modeled as

$$f_r(d) = \frac{-\kappa^2}{d}. \tag{5.4}$$

For each iteration of the algorithm, this force is applied between all pairs of vertices. The repulsive force decreases at larger distances, and its repulsive nature is reflected by the negative sign. Magnitude of the attractive force $f_a(d)$ between 2 vertices is modeled as

$$f_a(d) = \frac{d^2}{\kappa}. \tag{5.5}$$

This force is applied only between adjacent vertices (those connected by an edge of the original graph). It provides an attractive force that increases quadratically with distance. The directions of the forces between 2 vertices are in a direct line to the other vertex.

Figure 5-4 shows the attractive and repulsive forces corresponding to Eqs. (5.4) and (5.5), along with their sum. This combined force is

$$f_a(d) + f_r(d) = \frac{d^2}{\kappa} - \frac{\kappa^2}{d}, \tag{5.6}$$

which is zero at the ideal distance $d = \kappa$. Thus edge distances in the layout tend to converge toward κ under iterations of the algorithm. An example convergence of the spring algorithm is shown in Figure 5-5.

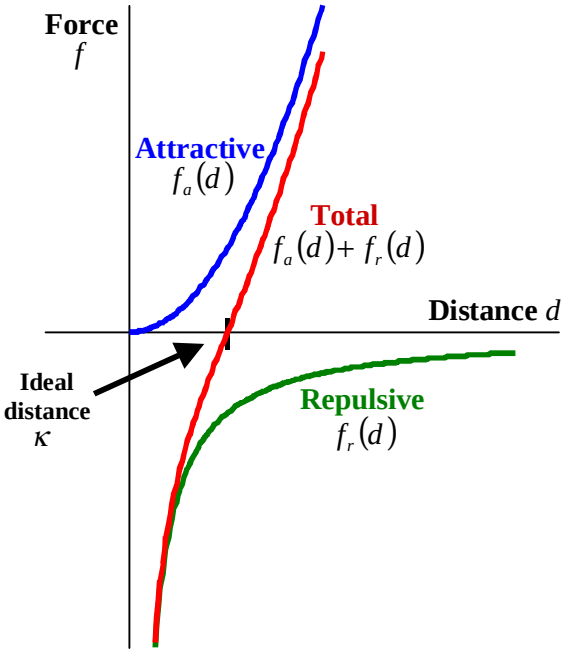


Figure 5-4: Attractive, repulsive, and total forces versus distance. Force cancel at ideal distance $d = \kappa$.

In Chapter 3, I described the application of Pearson's correlation in computing co-citation similarities. Correlations and co-citation counts are alike in that they are both dot products of a pair of vectors (or generalized dot product for higher-order co-citations). The difference is that for correlations, the vectors are first converted to z scores (zero mean and unit standard deviation). Co-citation count has a more direct interpretation, as the actual number of documents that co-cite a pair.

Another difference between co-citation counts and correlations is that for a given document set, counts tend to have more repeated distance values than do correlations. This is because counts are independent of mean and variance. You could interpret correlations as having finer distance granularity.

A consequence of this is that the minimum spanning trees for distances computed from correlations tend to be more "chained." That is, the average degree of a minimum

spanning tree vertex tends to be lower for correlation-based distances. This causes edges were formerly incident on "central" vertices of count-based minimum spanning trees to be "stretched out" in chains over multiple vertices. See the examples in Figures 5-6 through 5-8, for the SCI "Wavelets" data set.

A consequence is that count-based minimum spanning trees are less likely to become stuck in local optima during force-directed placement. This problem becomes more acute as the number of vertices increases. There are possible modifications for the placement algorithm to help avoid these local minima, such as simulated annealing, but the count-based minimum spanning trees are fundamentally easier to place. Count-based minimum spanning trees also make more efficient use of the placement area, in some sense "filling" more of the plane, in the sense of fractal geometry [Mand81].

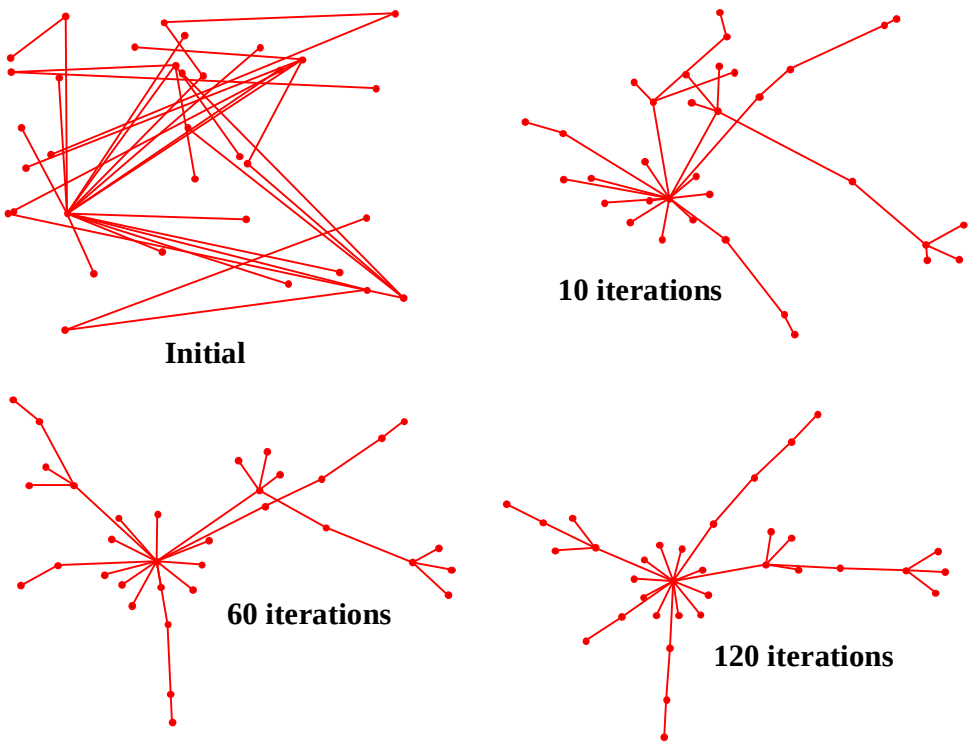
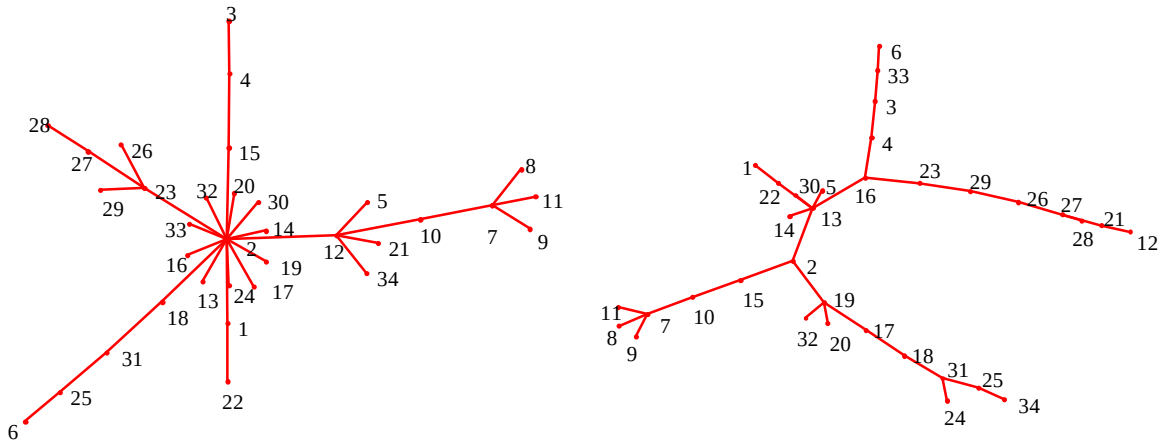
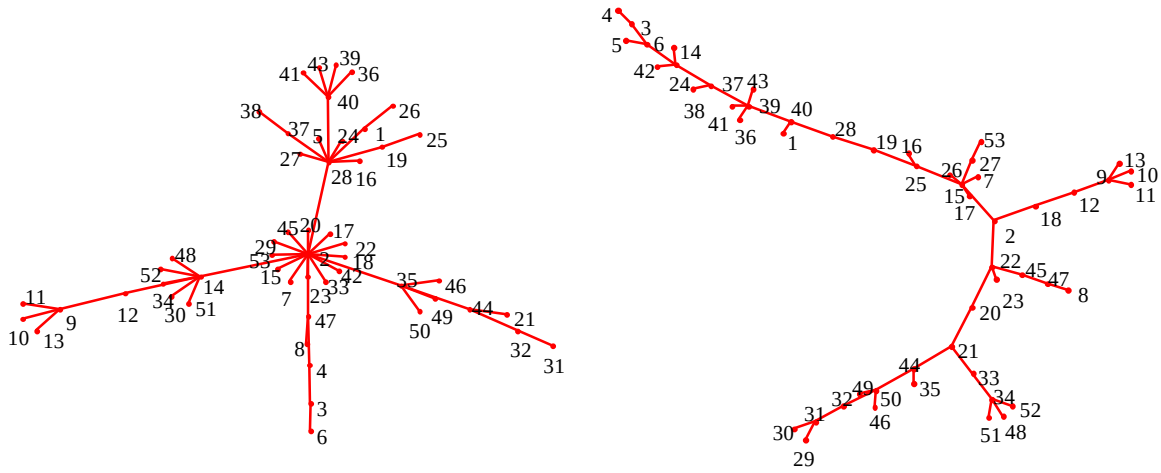


Figure 5-5: Iterations of spring algorithm for placing vertices of minimum spanning tree.



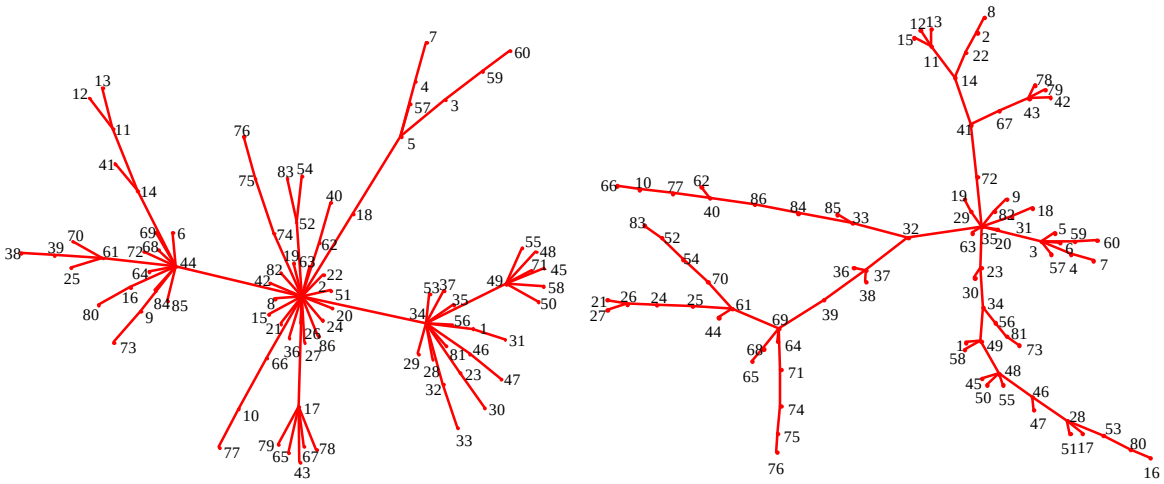
Co-Citation Count **Citation Correlation**

Figure 5-6: Minimum spanning tree placement for pairwise distances computed via co-citation count versus citation correlation, for data set "Wavelets 1999 (1-100)."



Co-Citation Count **Citation Correlation**

Figure 5-7: Minimum spanning tree placement for pairwise distances computed via co-citation count versus citation correlation, for data set "Wavelets 1999 (1-150)."



Co-Citation Count **Co-Citation Correlation**

Figure 5-8: Minimum spanning tree placement for pairwise distances computed via co-citation count versus citation correlation, for data set "Wavelets 1999 (1-200)."

This section described an algorithm for computing spatial coordinates for the minimum spanning tree, for the purpose of visualization. The next section introduces new metrics for measuring the effects of my new hybrid pairwise/higher-order distances on the minimum spanning tree, in terms of frequent itemsets.

5.3 Itemset-Matching Minimum Spanning Tree Metrics

Chen has applied the minimum spanning tree to co-citation analysis [Chen99a][Chen99b]. The interpretation is that the minimum spanning tree provides a network of influences among a collection of documents. The edges of the tree are considered to be essential links representing the most direct influences among documents. Documents near the center of the tree are interpreted as highly influential foundational

works, while those near the perimeter represent the emerging research front. This is illustrated in Figure 5-9.

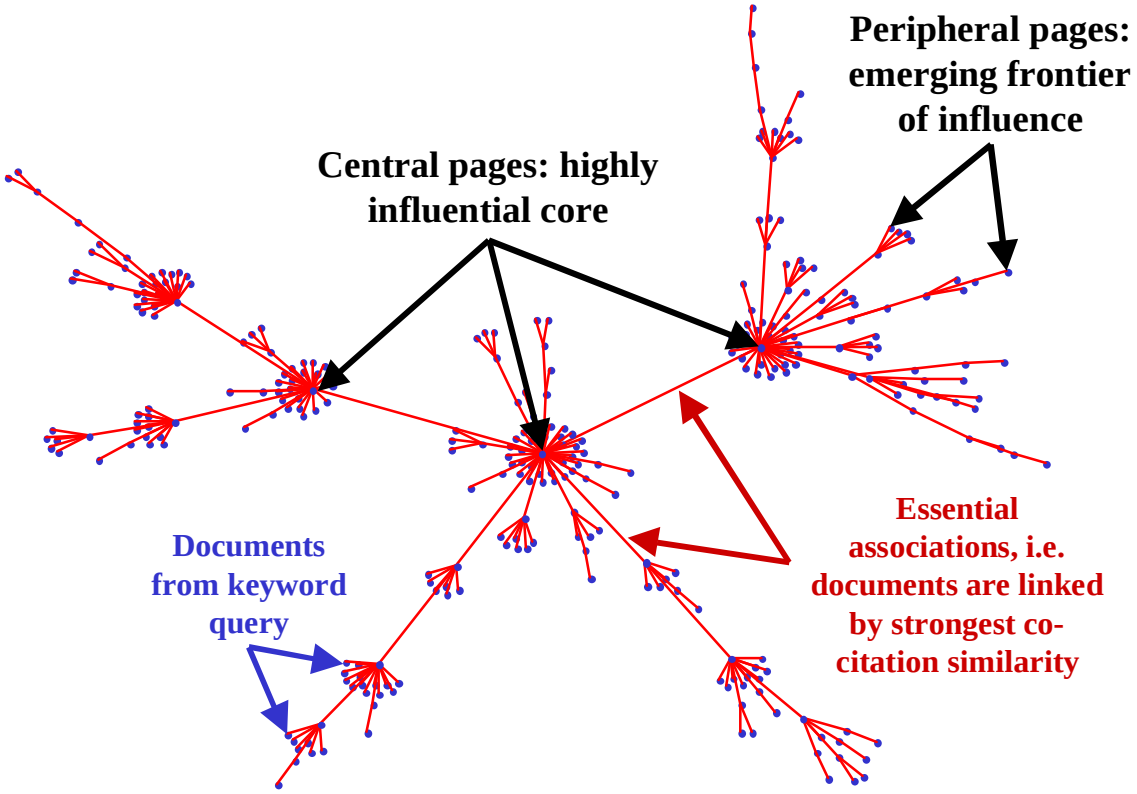


Figure 5-9: Minimum spanning tree visualization serves as network of document influences.

The minimum spanning tree as a network of influence is illustrated in Figure 5-10 for the SCI "Wavelets (1-200)" data set. Documents recognized from the network as highly influential have the author and publication year identified. These particular authors would be widely regarded as being highly influential in the field of wavelets.

In this section, I examine the effect of my proposed hybrid pairwise/higher-order distances on the minimum spanning tree influence network visualization. In general, the new distances generate networks having more direct influences among members of

frequent itemsets. The hybrid distances also tend to draw frequent itemset documents closer to the network center, increasing their relative influence of within the network.

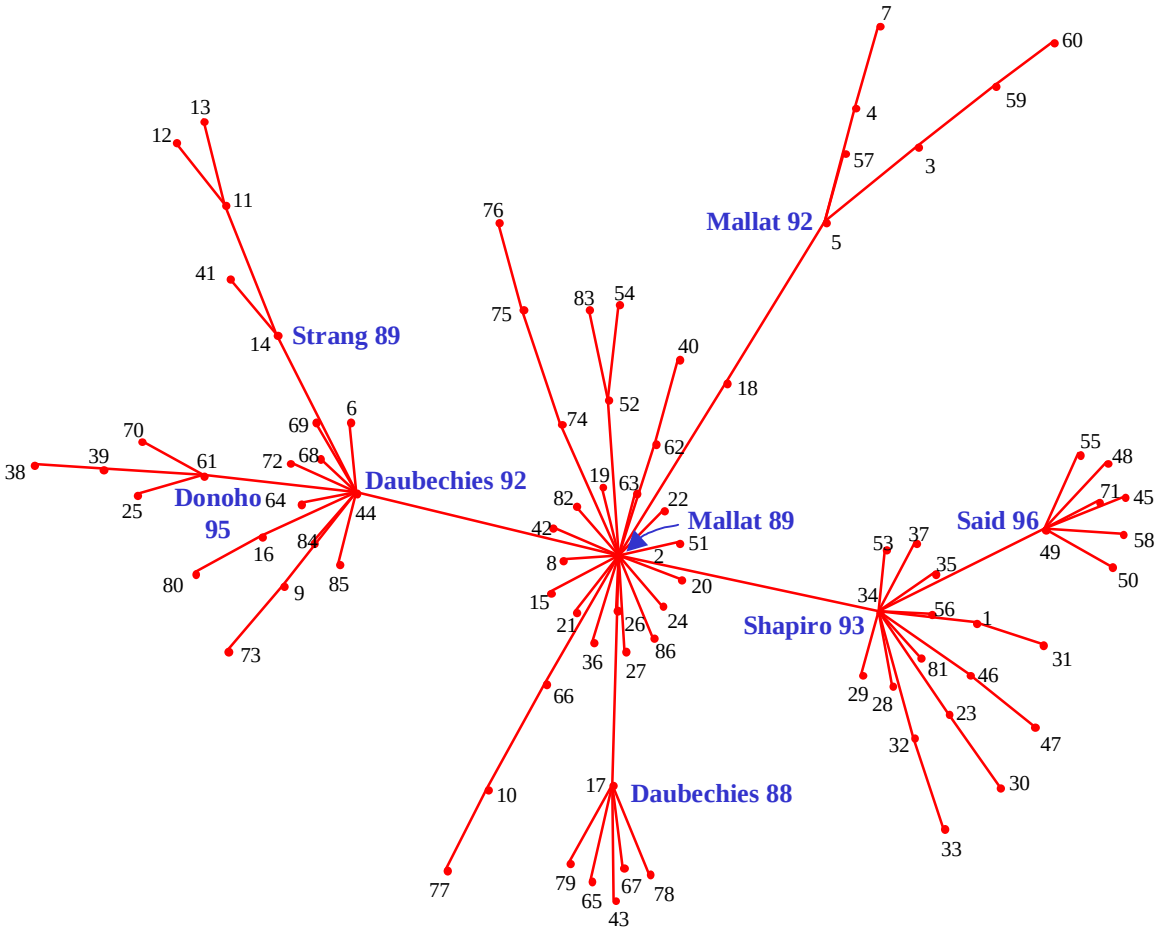


Figure 5-10: Minimum spanning tree network of influence for wavelets documents cited in 1999. Identified documents are generally recognized as highly influential.

In particular, this section proposes 3 new itemset-based metrics for evaluating the effect of hybrid pairwise/higher-order distances on the minimum spanning tree network. One metric measures the extent to which frequent itemset members directly influence one another, by the number of connected components they form in the tree. The other 2

metrics assess the relative influence of frequent itemset members within the network. One measures influence for a single member by its graph-theoretic degree, indicating the number of other documents it influences. The other assesses the influence of frequent itemsets as a whole within the entire network.

There are some basic ideas that are helpful in understanding the effect of distance formulas on the minimum spanning tree. These ideas are better understood through a comparison of the minimum spanning tree and graph-theoretical clustering.

Single-linkage clustering for a given clustering threshold is equivalent to the connected components of the minimum spanning tree after the removal of edges larger than that threshold. Like the minimum spanning tree, the dendrogram is also a tree, of threshold-dependent merges. While the dendrogram tree shows the threshold at which a document is merged into a cluster, it does not show the exact nearby document that caused the merge. This is precisely what the minimum spanning tree shows.

Moreover, documents with smaller distances among them appear near the leaves of the dendrogram tree, because clustering agglomerates from the bottom up. These documents are the highly influential ones that form the central core of the minimum spanning tree. Figure 5-11 shows this for the SCI “Wavelets (1-200)” data set. The figure identifies documents at the 3 lowest levels of the single-linkage dendrogram. These same documents are also identified in the dendrograms for complete and average linkage, as well as in the minimum spanning tree.

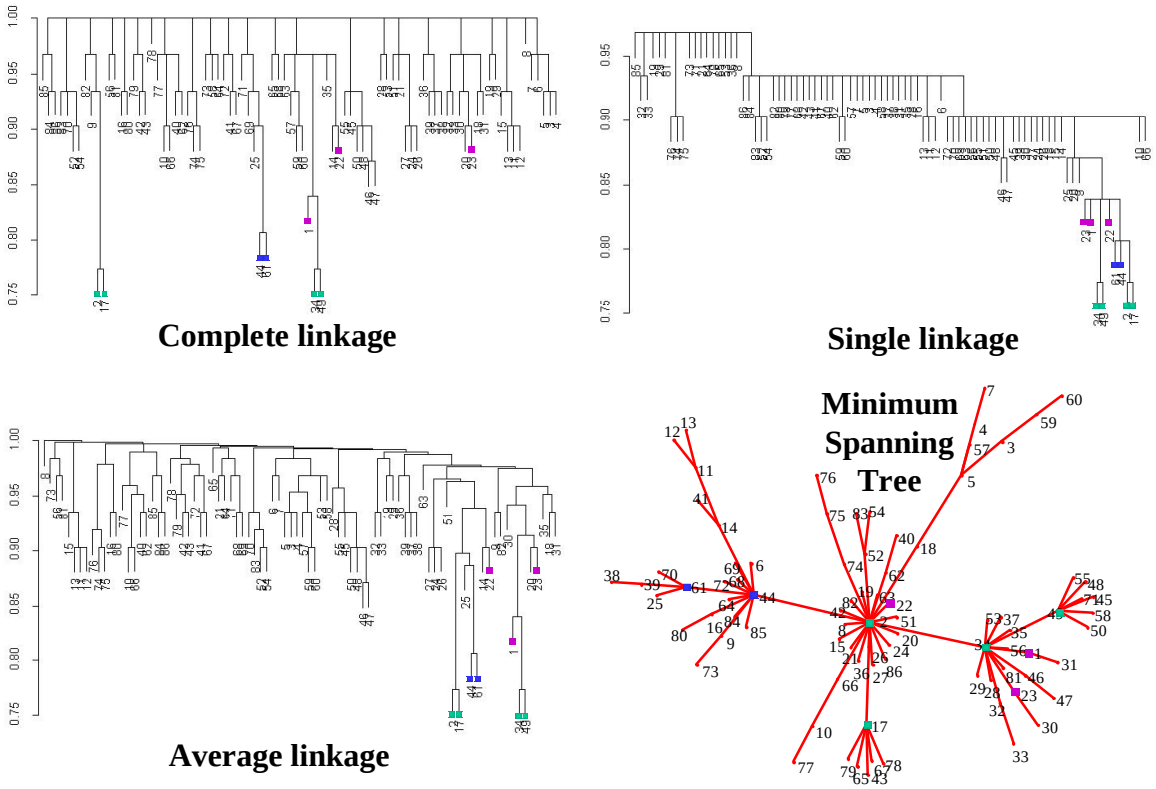


Figure 5-11: Documents at the lower levels of the single-linkage dendrogram tend to be near the center of the minimum spanning tree.

I now show how these ideas apply in terms of documents in frequent itemsets, for standard pairwise versus hybrid pairwise/higher-order distances. This is shown in Figures 5-12 through 5-17. The intention is to gain insight into exactly how to construct a metric for evaluating distance formulas in terms of frequent itemsets.

Figures 5-12 through 5-14 are for the SCI "Wavelets 1999 (1-100)" data set. They compare clustering, the minimum spanning tree, and frequent itemsets of cardinality 4. Distances are computed in 3 ways: (1) standard pairwise (cardinality-2 itemset supports), (2) from summation of cardinality-4 itemset supports raised to the 4^{th} power, and (3) from summation of 4^{th}-power supports for cardinalities 2, 3, and 4.

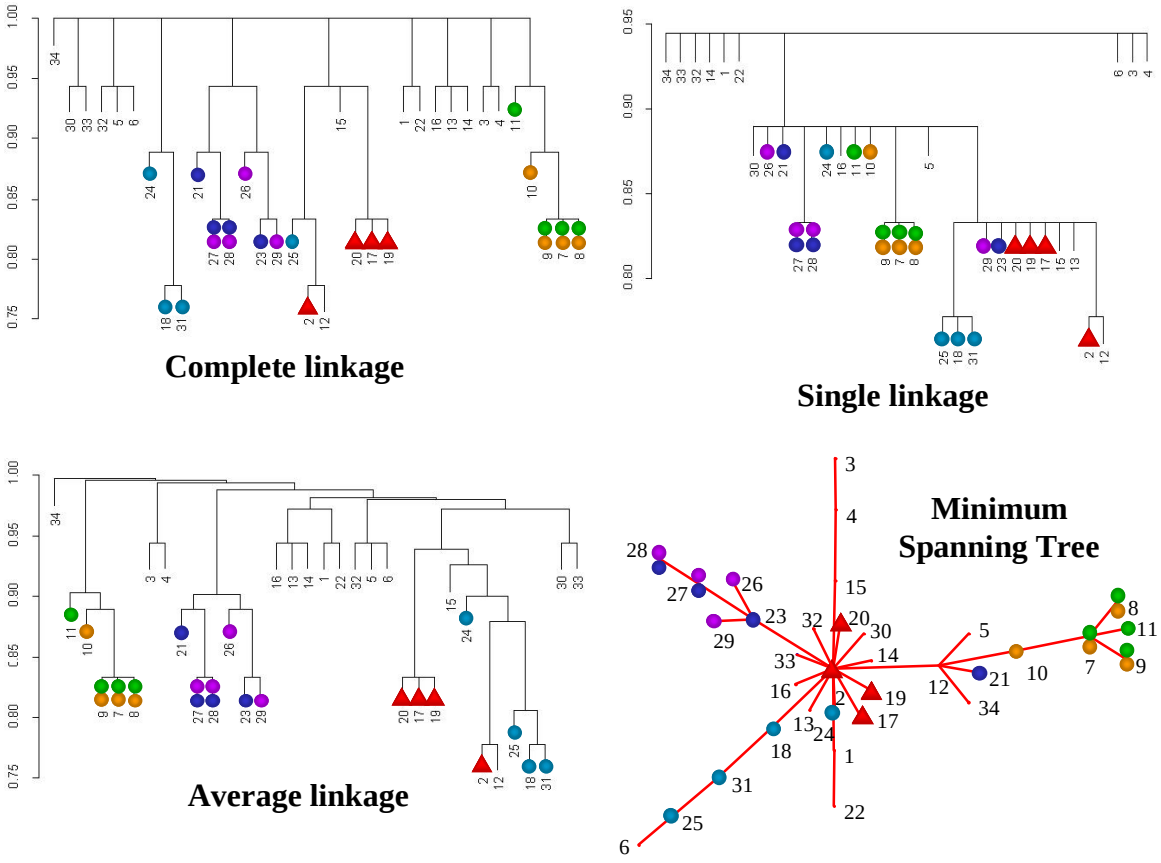


Figure 5-12: Comparison of clustering, frequent itemsets, and minimum spanning tree for data set "Wavelets 1999 (1-100)" with pairwise distances.

Chapter 3 showed that frequent itemsets tend to be consistent with graph-theoretic clusters. Consistency is increased for the hybrid pairwise/higher-order distances I proposed. I now compare frequent itemsets to minimum spanning trees. In general, frequent itemsets and the edges between them tend to form connected components of the minimum spanning tree. I hypothesize that this tendency increases for hybrid pairwise/higher-order distances.

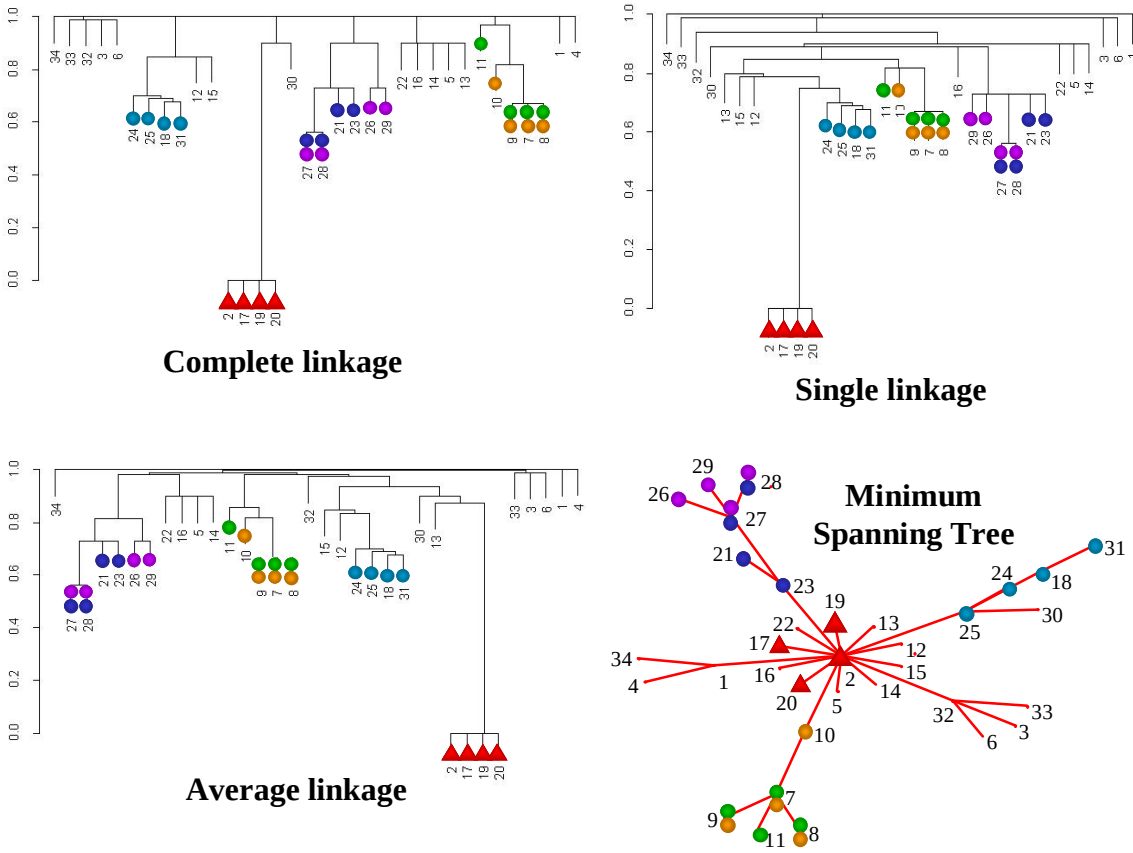


Figure 5-13: Comparison of clustering, frequent itemsets, and minimum spanning tree for data set "Wavelets 1999 (1-100)" with distances from order-4 co-citations.

In this particular example, for pairwise distances only 3 out of 6 of the frequent itemsets form minimum spanning tree connected components: {2, 17, 19, 20}▲, {7, 8, 9, 10}●, and {7, 8, 9, 11}●. Two of the 6 frequent itemsets each form pairs of connected components: {24}● and {18, 25, 31}●, and {21}● and {23, 27, 28}●. The remaining frequent itemset forms 3 connected components: {27, 28}●, {26}●, and {29}●.

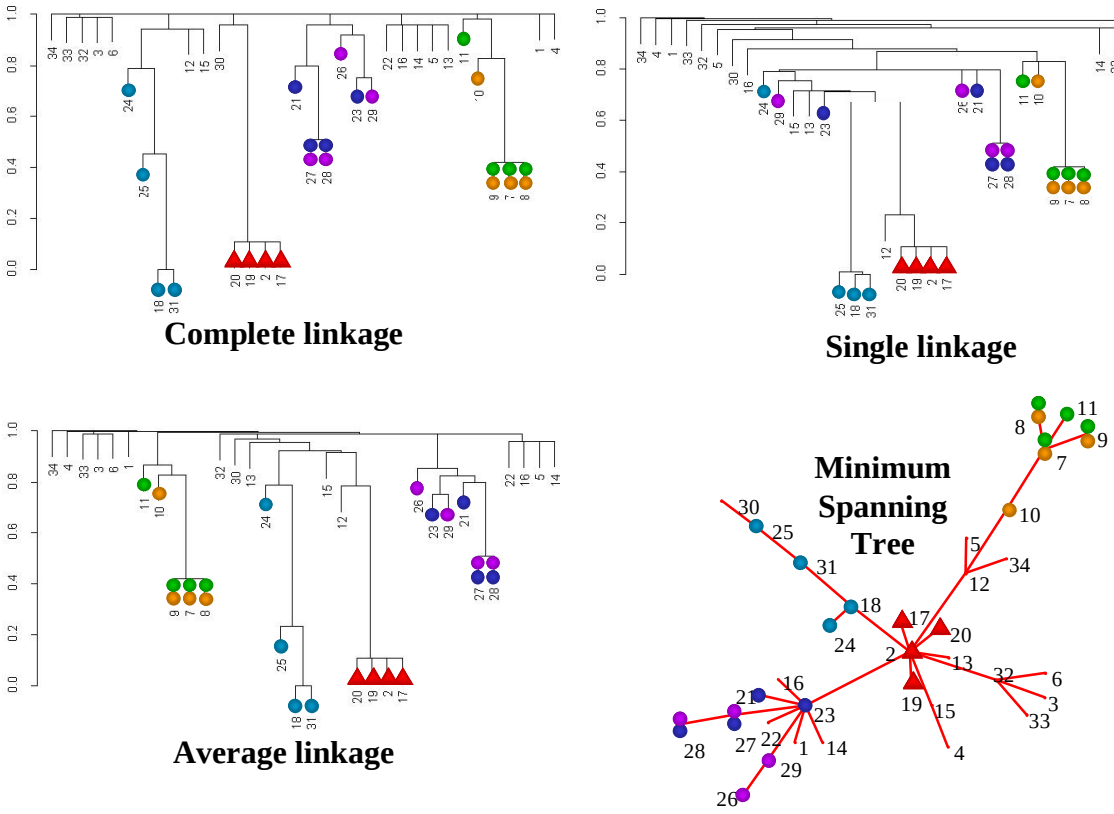


Figure 5-14: Comparison of clustering, frequent itemsets, and minimum spanning tree for data set "Wavelets 1999 (1-100)," distances from co-citations of orders 2, 3, 4.

For the hybrid pairwise/higher-order distances, almost all 6 of the large itemsets form single minimum spanning tree connected components. In particular, for hybrid distances computed from only cardinality-4 itemsets, all frequent itemsets form connected components. For hybrid distances computed from itemsets of cardinalities 2,3,4, all frequent itemsets form connected components with the exception of one that forms 2 connected components: {26, 29}● and {27, 28}●.

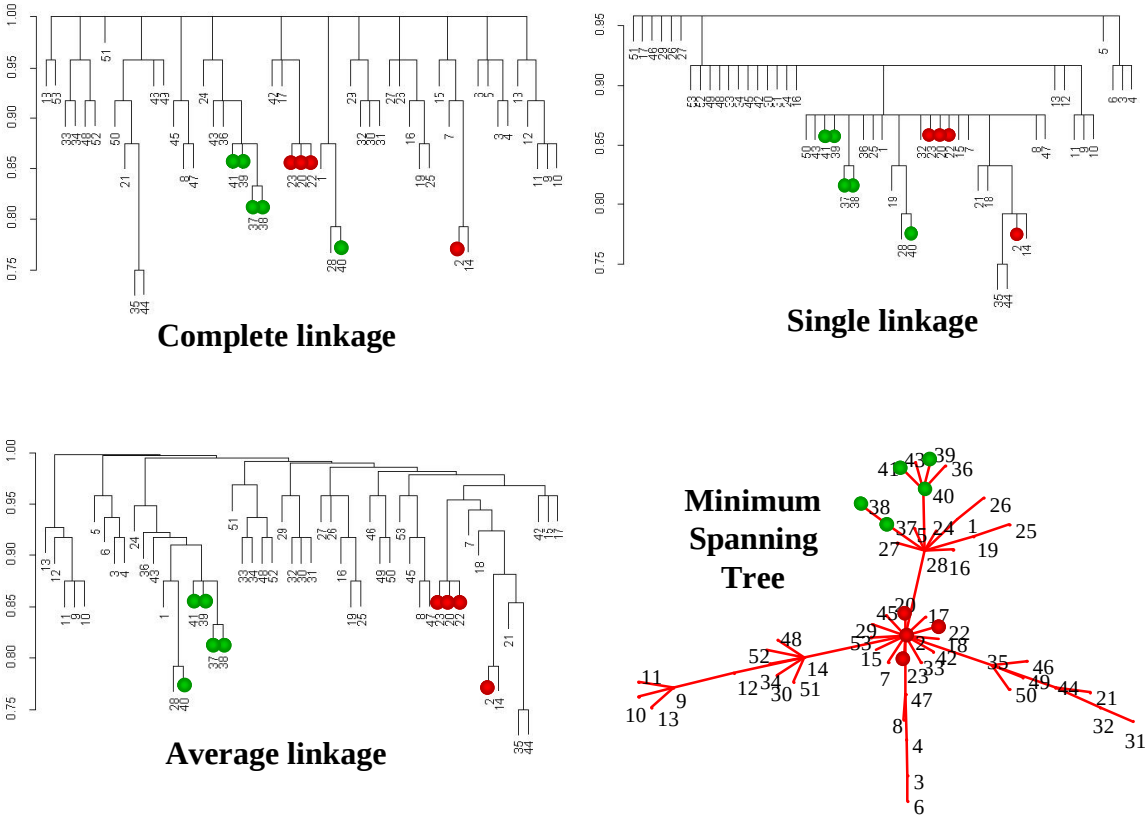


Figure 5-15: Comparison of clustering, frequent itemsets, and minimum spanning tree for data set "Wavelets 1999 (1-150)" with pairwise distances.

Note that the cardinality-4 frequent itemsets comparison form fewer connected components for the distances computed from cardinality-4 itemsets only. That is, the cardinality of the distance itemsets is the same as the cardinality of frequent itemsets that we are testing for being connected. This is consistent with the idea that computing similarities from nonlinearly transformed frequent itemsets corresponds to decreasing the corresponding distances among their items. These smaller distances are more likely to connect the items in the tree, since the minimum spanning tree is composed of minimum-distance edges.

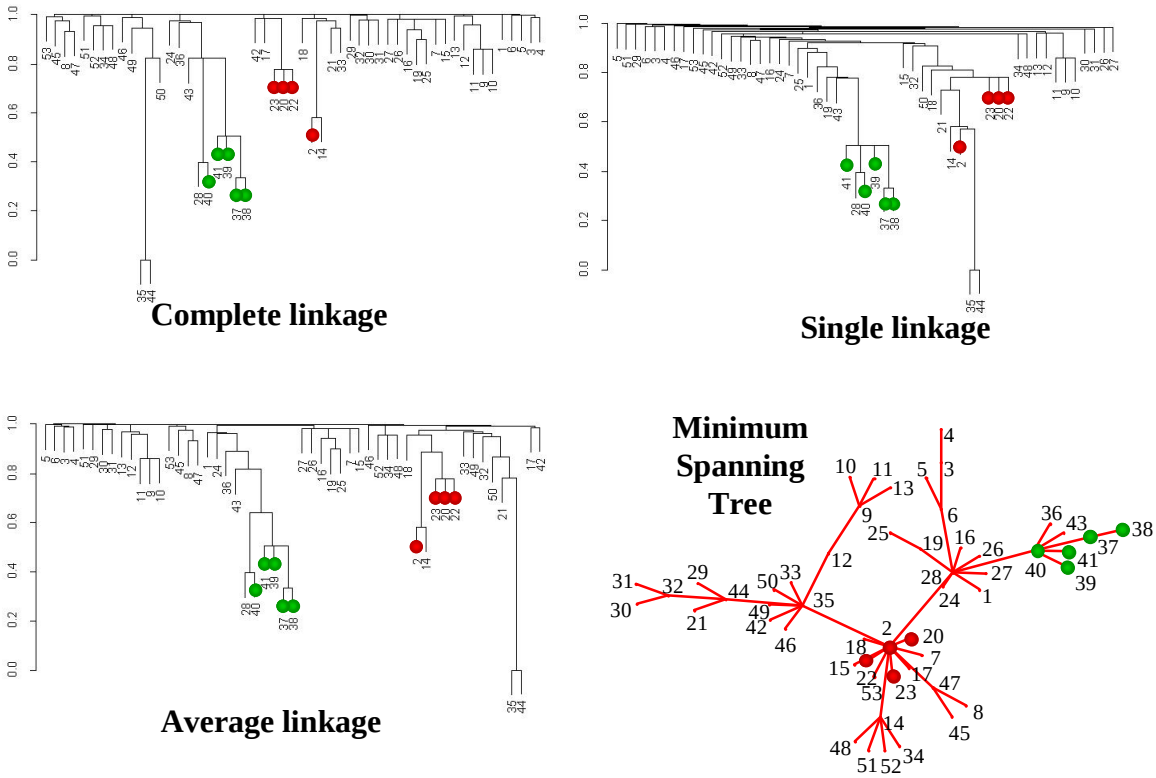


Figure 5-16: Comparison of clustering, frequent itemsets, and minimum spanning tree for data set "Wavelets 1999 (1-150)" with distances from higher-order co-citations of cardinalities 2, 3, 4.

Figures 5-15 through 5-17 demonstrate these ideas for a larger data set, i.e. "Wavelets 1999 (1-150)" data set. Clusterings and the minimum spanning tree are compared to the cardinality-4 frequent itemset {2, 20, 22, 23}● and the cardinality-5 frequent itemset {37, 38, 39, 40, 41}●. Results for standard pairwise distances are similar as for the smaller data set. In particular, for the minimum spanning tree the frequent itemset {2, 20, 22, 23}● is connected, and the frequent itemset {37, 38, 39, 40, 41}● is divided into 2 connected components.

For hybrid pairwise/higher-order distances in Figure 5-15, the 2 frequent itemsets are connected in the minimum spanning tree. However, they do not form clusters. I

therefore removed the cardinality-2 supports from the distance computation. That is, rather than compute distances from 4th power supports summed over cardinalities 2 through 4, I compute them over cardinalities 3 and 4 only. With these new distances, the 2 itemsets are consistent with the resulting clusters.

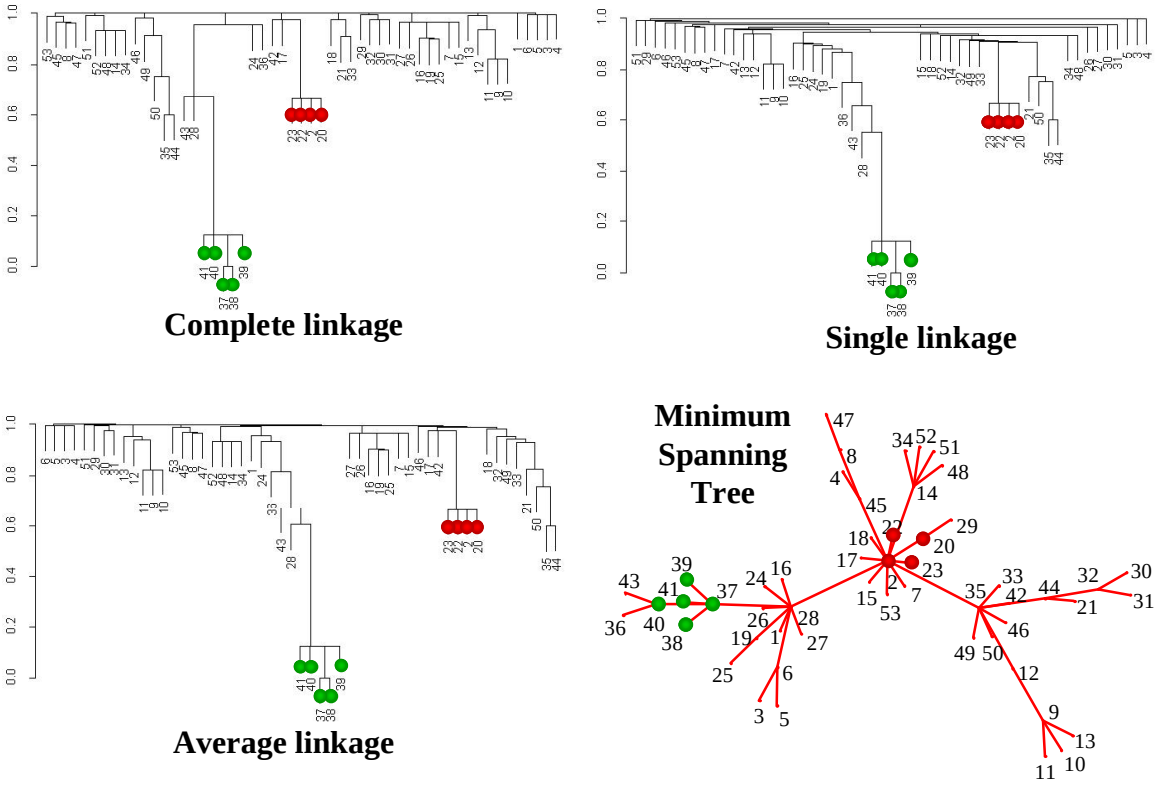


Figure 5-17: Comparison of clustering, frequent itemsets, and minimum spanning tree for data set "Wavelets 1999 (1-150)" with distances from higher-order co-citations of cardinalities 3, 4.

In the examples I have shown, frequent itemsets (or more precisely the edges between them) tend to form connected subsets of the minimum spanning tree. This tendency increases as itemset supports are nonlinearly transformed. It also increases when itemsets cardinalities used for distances match those used in the metric.

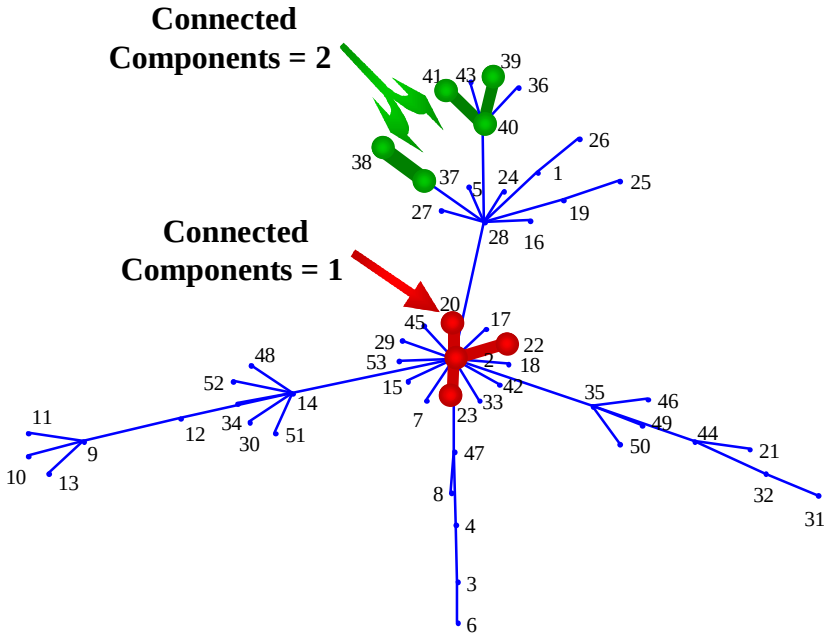


Figure 5-18: Number of connected components of frequent itemsets forms basis for itemset-matching minimum spanning tree metric.

This suggests a new metric for comparing minimum spanning trees resulting from various distance formulas. The metric is the inverse of the average number of minimum spanning tree connected components for a frequent itemset. Figure 5-18 illustrates numbers of itemset connected components of the minimum spanning tree.

Consider the minimum spanning tree $MST(G) = (V, E_t)$ of the graph $G = (V, E)$, with minimum spanning tree edges $E_t = (u_t, v_t)$. Here G is the undirected, weighted, fully connected graph corresponding to the distance matrix.

Consider also an itemset $I_j \subseteq V$. For a given minimum spanning tree (V, E_t) and itemset I_j, there is some set of edges E_j that are each incident on exactly 2 vertices

in I_j, i.e. $E_j = \{E_t | u_t, v_t \in I_j\}$. By a property of trees, the number of connected components c_j for itemset I_j is

$$c_j = |I_j| - |E_j|. \tag{5.7}$$

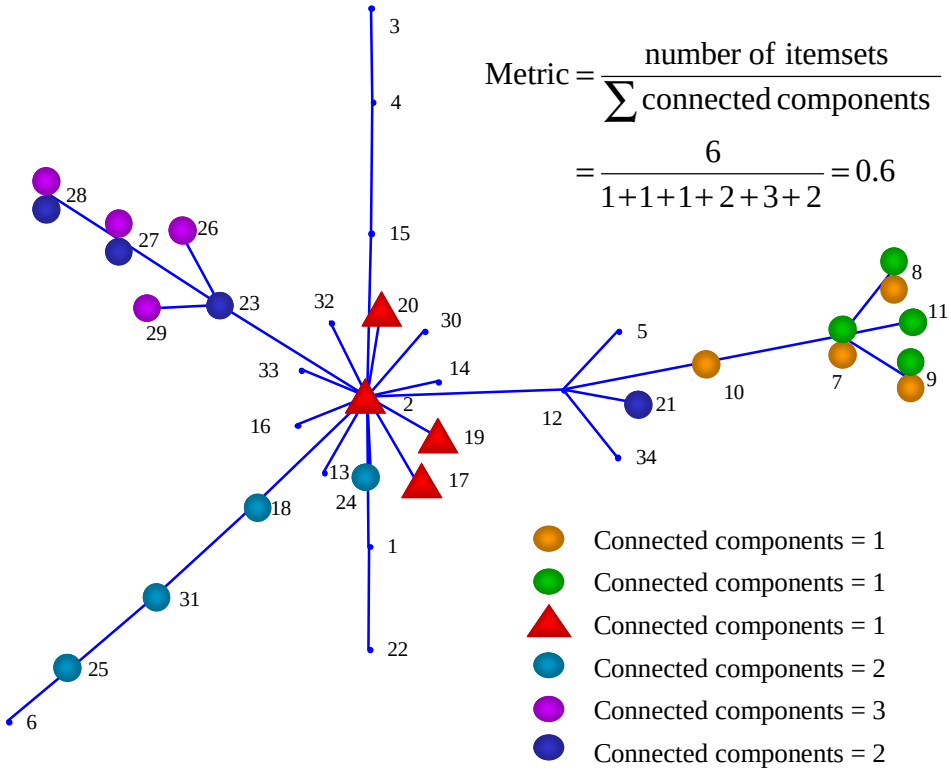


Figure 5-19: Metric for minimum spanning tree is inverse of average number of connected components per itemset.

The metric value $m_{CC}(I)$ for a set of itemsets $I = \{I_1, I_2, \ldots, I_j, \ldots, I_{|I|}\}$, with respect to minimum spanning tree $MST(G)$, is then

$$m_{CC}(I) = |I| \left(\sum_{j=1}^{|I|} c_j \right)^{-1}. \tag{5.8}$$

The summation of numbers of connected components is in the denominator so as to make metric values smaller for larger numbers of connected components. The metric takes its maximum value of unity when $c_j = 1$ for all c_j. The inverse of the metric $m_{\text{CC}}^{-1}(I)$ is the average number of connected components per itemset. Figure 5-19 illustrates the metric.

The theoretical guarantee I provided in Chapter 3 for the matching of frequent itemsets to clusters does not directly apply here, since it was developed for disjoint clusters. But the spirit of theorem applies. In particular, there is always some sufficiently large degree of nonlinearity in the itemset support transformation such that more frequent itemsets will not be divided into multiple connected components via less frequent itemsets. As before, the guarantee provides no upper bound on the necessary degree of nonlinearity.

In comparison to the clustering metric I introduced in Chapter 3, this new connected components based metric for the minimum spanning tree can be considered a more stringent requirement for testing the effect of distances on itemsets. For example, consider a cluster that contains only members of some itemset, with at least one itemset member not already in the cluster. The merging into the cluster of a document outside the itemset, at the expense of an itemset member, requires only one closer document distance. But the separation of an itemset member on the minimum spanning tree requires that there be no link from that member to another member.

This is a stronger method of keeping frequent itemsets in a single set, since competition from documents outside the itemset need not interfere. In other words, for the minimum spanning tree connected-components criterion, it is more difficult to divide itemset members into multiple sets. Any method that results in separation of itemsets

into multiple connected components has an even stronger ability to prevent exclusive clustering of those itemsets. These ideas are supported by the empirical results in the next section.

From the examples I have shown in this section, it is evident that vertices near the center of the minimum spanning tree network visualization often have relatively high graph theoretic degrees. That is, they often have larger numbers of edges incident on them. Since hybrid pairwise/higher-order distances tend to draw frequent itemset members closer to the center of the network, this suggests the graph-theoretic degree as the basis for a different metric. In particular, the metric can average the graph-theoretic degree over a number of itemsets.

Consider the minimum spanning tree $MST(G) = (V, E_t)$ of the distance graph $G = (V, E)$, with minimum spanning tree edges $E_t = (u_t, v_t)$. Consider also an itemset $I_j \subseteq V$. Each vertex $v_i \in I_j$ has degree $\delta(v_i)$ with respect to edges E_t. The average degree over all vertices in I_j is

$$ave(I_j) = \frac{1}{|I_j|} \sum_{v_i \in I_j} \delta(v_i). \tag{5.9}$$

Thus the metric value $m_{\mathrm{D}}(I)$ for a set of itemsets $\{I | I_j \in I\}$ with respect to $MST(G)$ is

$$\begin{aligned} m_{\mathrm{D}}(I) &= \frac{1}{|I|} \sum_{I_j \in I} ave(I_j) \\ &= \frac{1}{|I|} \sum_{I_j \in I} \left[|I_j|^{-1} \sum_{v_i \in I_j} \delta(v_i) \right]. \end{aligned} \tag{5.10}$$

When itemset cardinality $\chi = |I_j|$ is fixed for all j, the metric becomes

$$m_{\mathrm{D}}(I) = \frac{1}{\chi|I|} \sum_{I_j \in I} \sum_{v_i \in I_j} \delta(v_i) \,. \tag{5.11}$$

A 3rd metric is possible by the examination of influences of itemset members in the entire minimum spanning tree visualization network. This is in terms of numbers of descendents in the tree for frequent itemset members. For a given assigned root of the tree, the metric for a single itemset member is its number of descendents with respect to the root. The metric could then be averaged over all members of an itemset, and again over a number of itemsets. This yields the average number of descendents for an itemset member, for a given set of frequent itemsets, for a given minimum spanning tree.

It is generally convenient to leave the tree unrooted, which allows the user flexibility in interpreting the tree visualization. But a root must be assigned for this itemset-descendent metric.

A root is arbitrarily assigned in actually computing the minimum spanning tree, as the first vertex that encountered. But this is clearly not useful for our purpose of interpreting the minimum spanning tree as a network of document influences. Here it is better to assign a root document based on strength of influence. That is, the document with the strongest influence within the network should be the root. Possibilities include the document with largest vertex degree, the document at the graph-theoretic center of the tree, *a priori* information from the user, or some combination of these.

This section has proposed metrics for measuring the effects of various distance formulas on the minimum spanning tree. In the next section, I apply these metrics to data sets extracted from a citation database, in order to test the effects of my new hybrid pairwise/higher-order distances.

5.4 Minimum Spanning Tree Experiments

The previous section described 3 itemset-based metrics for assessing the impact of various distance formulas on the minimum spanning tree network-of-influence visualization. These metrics support analyses of the influences of frequent itemset members, either among the members themselves or within the entire network. This section applies the metrics to various SCI (Science Citation Index) data sets, to test the effects of hybrid pairwise/higher-order distances on the minimum spanning tree visualization. This includes testing the effects of techniques for reducing the complexity of distance computations.

In Section 5.3, I proposed a metric $m_{\mathrm{CC}}(I)$ for determining the extent that documents in frequent itemsets I are connected in the minimum spanning tree $MST(G)$:

$$m_{\mathrm{CC}}(I) = |I| \left(\sum_{j=1}^{|I|} c_j \right)^{-1}. \tag{5.12}$$

Here $MST(G) = (V, E_t)$ is the minimum spanning tree of the fully connected distance graph $G = (V, E)$, with minimum spanning tree edges $E_t = (u_t, v_t)$. Also, $c_j = |I_j| - |E_j|$ is the number of connected components in $MST(G)$ for itemset $I_j \subseteq I$, and $E_j = \{E_t | u_t, v_t \in I_j\}$ is some set of edges that are each incident on exactly 2 vertices in itemset I_j. The metric takes its maximum value of unity when the number of connected components $c_j = 1$ for all c_j, indicating maximum itemset connectedness. The inverse of the metric $m_{\mathrm{CC}}^{-1}(I)$ is the average number of connected components per itemset.

I now apply the minimum spanning tree itemset-connectedness metric $m_{\mathrm{CC}}(I)$ to the SCI data sets described in Table 5-1. Results are included for both co-citations and bibliographic coupling for the data sets "Adaptive Optics," "Quantum Gravity and Strings," and "Wavelets and Brownian," so that there are 10 total data sets. I compare metric values between standard pairwise distances and the new hybrid pairwise/higher-order distances. This includes a comparison of metrics resulting from hybrid distances with reduced computational complexity.

Table 5-1: Details for SCI data sets used in this section. Bibliographic coupling is applied in addition to co-citations for 3 of these data sets.

Data set name	Query keyword	Year(s)	Citing docs	Cited docs
Adaptive Optics	adaptive optics	2000	89	60
Collagen	collagen	1975	494	53
Genetic Algorithms and Neural Networks	genetic algorithm* and neural network*	2000	136	57
Quantum Gravity and Strings	quantum gravity AND string*	1999-2000	114	50
Wavelets (1-100)	wavelet*	1999	100	34
Wavelets (1-500)	wavelet*	1999	472	54
Wavelets and Brownian	wavelet* AND brownian	1973-2000	99	59

Standard pairwise document similarities are computed as in Eq. (5.5), and hybrid similarities as in Eq. (5.6). Similarities with reduced complexity are computed by excluding itemset supports below *minsup*, as described by Eq. (5.9). Because of the poor performances of transaction and item weighting in the previous chapter, these complexity reduction techniques are excluded here. Each type of document similarity is normalized via Eq. (5.7), and converted from similarity to distance by linear inversion, i.e. Eq. (5.8).

Tables D-1 through D-10 in Appendix D show the values of the itemset-connectedness metric for the SCI data sets. In the tables, distances are denoted either pairwise or of the form o_χ^p, which indicates the summation of itemset supports raised to the power p, over itemsets of cardinality χ. The tables also indicate the k most frequent itemsets that are included in the metric, the metric itemset cardinality χ, denoted o_χ, and any value of *minsup* that is applied.

Just as for clustering metrics in the previous chapter, the cardinality χ is consistent in each test case between distances o_χ^p and frequent itemsets o_χ, with the exception of pairwise distances o_2^1. Such consistency allows a more direct interpretation of the metric. Also, sets of test cases with identical itemsets applied in the metric are marked with alternating black and red text, which aids in the comparison of standard and hybrid distances.

Tables 5-2 through 5-4 summarize the minimum spanning tree itemset-connectedness results in Tables D-1 through D-10. Table 5-2 includes only cases having no application of *minsup* for reducing complexity, while Tables 5-3 and 5-4 correspond to values of *minsup* = 2 and *minsup* = 4, respectively.

The itemset-connectedness metric for minimum spanning trees has much less distance-dependent variation compared to the itemset-matching clustering metric. For the minimum spanning tree metric, values changed between standard pairwise and hybrid pairwise/higher order distances in only about 25% of the test cases. This supports my characterization in the previous section of the minimum spanning tree itemset-connectedness metric as a more stringent test for the effect of distances on itemset

members. That is, it is more difficult to make itemset members disconnected in the minimum spanning tree than it is to prevent their exclusive clustering.

Table 5-2: Minimum spanning tree itemset-connectedness metric for standard pairwise (P.W.) versus hybrid (H.O.) distances.

Data set	H.O.=P.W.	H.O.>P.W.	H.O.<P.W.	Cases
1	8	4	0	12
2	3	3	3	9
3	6	0	0	6
4	8	1	0	9
5	6	0	0	6
6	8	0	1	9
7	3	3	0	6
8	6	0	0	6
9	6	0	0	6
10	2	4	0	6
Totals	56	15	4	75

Table 5-3: Comparisons of minimum spanning tree itemset-connectedness metric for hybrid distances with full complexity (*minsup* 0) versus reduced complexity (*minsup* 2).

Data set	(*minsup* 2) = (*minsup* 0)	(*minsup* 2) > (*minsup* 0)	(*minsup* 2) < (*minsup* 0)	Cases
1	4	0	2	6
2	6	0	0	6
3	6	0	0	6
4	6	0	0	6
5	6	0	0	6
Totals	28	0	2	30

Table 5-4: Comparisons of minimum spanning tree itemset-connectedness metric for hybrid distances with full complexity (*minsup* 0) versus reduced complexity (*minsup* 4).

Data set	(*minsup* 4) = (*minsup* 0)	(*minsup* 4) > (*minsup* 0)	(*minsup* 4) < (*minsup* 0)	Cases
1	4	0	2	6
2	6	0	0	6
3	6	0	0	6
4	6	0	0	6
5	6	0	0	6
Totals	28	0	2	30

Still, for the 25% of the cases with differing metric values (19 out of 60 total), itemset-connectedness metric values were larger for the new hybrid distances almost 4 times as often as for standard distances (15 versus 4 cases, respectively). Of the 60 cases with varying values of *minsup*, there were only 4 cases in which a nonzero *minsup* lowers the itemset-connectedness metric. I conclude that overall, despite the reduced variability, results for this metric show that frequent itemset members are generally more connected in the minimum spanning tree when the hybrid pairwise/higher-order distances are applied.

In the previous section, I proposed another itemset-based metric for minimum spanning trees. It measures the extent of direct influences that itemset members have in the tree, based on their graph-theoretic degree. The metric $m_{\mathrm{D}}(I)$ for a set of itemsets I with respect to minimum spanning tree $MST(G)=(V,E_t)$ of the distance graph $G=(V,E)$, with tree edges $E_t=(u_t,v_t)$ is

$$m_{\mathrm{D}}(I)=\frac{1}{\chi|I|}\sum_{I_j\in I}\sum_{v_i\in I_j}\delta(v_i) \tag{5.13}$$

Here, each vertex $v_i \in I_j$ of itemset $I_j \in I$ has graph-theoretic degree $\delta(v_i)$ with respect to edges E_t. Thus the metric $m_{\mathrm{D}}(I)$ is the average vertex degree over a number of itemsets.

The principal behind this metric is that vertices near the center of the minimum spanning tree network visualization often have higher degrees. The hypothesis is that hybrid pairwise/higher-order distances tend to draw frequent itemset vertices closer to the center of the network, thus tending to raise their degrees.

However, the results in Figures 5-13 through 5-22 only weakly support this hypothesis. Metric values for itemset members do generally increase as the itemsets become more frequent on average, i.e. as fewer frequent itemsets are included in the metric. This supports the idea that members of more frequent itemsets have larger vertex degrees in general.

But metric values are higher for hybrid pairwise/higher-order distances in only 60% of the test cases (12 out of 20). Of these 12 cases, 8 cases have conclusively higher metrics, and the remaining 4 are only marginally higher. Of the other 40% of the cases (8 out of 20), 4 cases have conclusively lower metrics for hybrid distances, and the remaining 4 are only marginally lower. In making these assessments I give higher priority to metric values for lower numbers of frequent itemsets. Given the interpretation that a larger vertex degree implies a larger number of direct influences on documents, the conclusion is that hybrid pairwise/higher-order distances only weakly increase such influence.

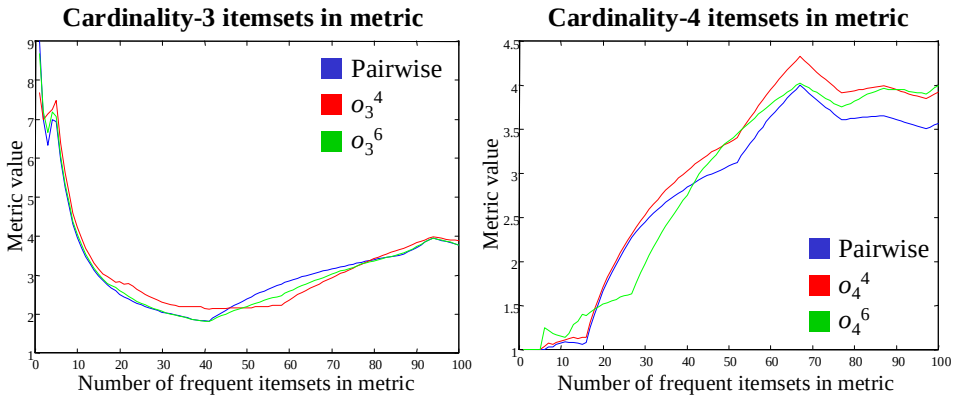


Figure 5-20: Minimum spanning tree itemset-degree metrics for "Adaptive Optics" data set.

In the previous section, I proposed a 3rd minimum spanning tree metric, which assesses the influence of frequent itemset members within in the minimum spanning tree visualization. It is defined in terms of tree descendents of itemset members, for a given root of the minimum spanning tree. The previous section describes an overall influence metric for a set of itemsets by averaging metric values over all itemset members and itemsets.

Rather than compute such an average metric, here I take a different approach. For the single most frequent itemset, I directly compare relative influences for each member, between standard pairwise and hybrid pairwise/higher-order distances. That is, I compute the influence metric for each itemset member of the itemset, and compare these individual metric values between distance formulas.

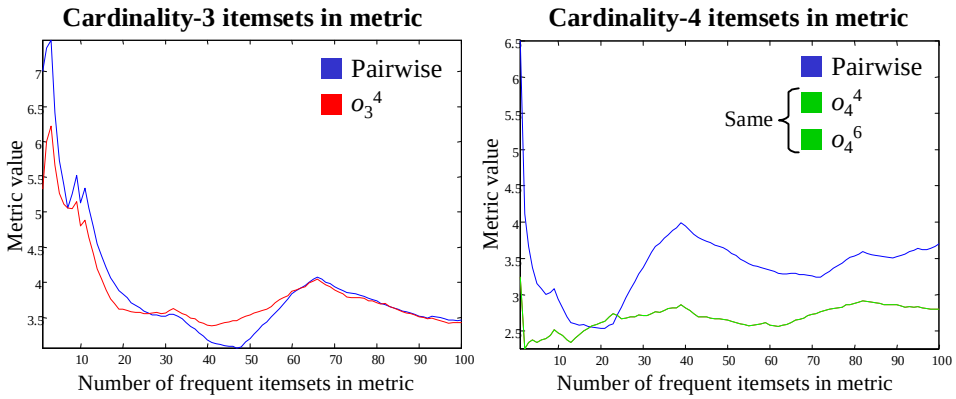


Figure 5-21: Minimum spanning tree itemset-degree metrics for "Adaptive Optics" data set, with bibliographic coupling.

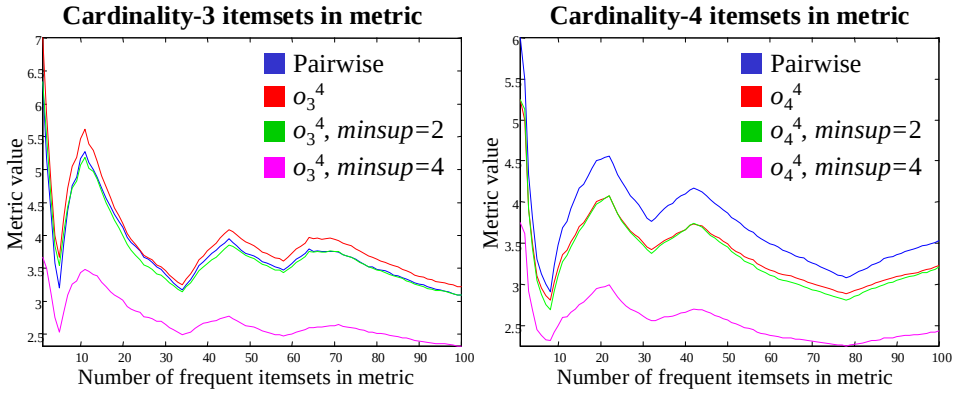


Figure 5-22: Minimum spanning tree itemset-degree metrics for "Collagen" data set.

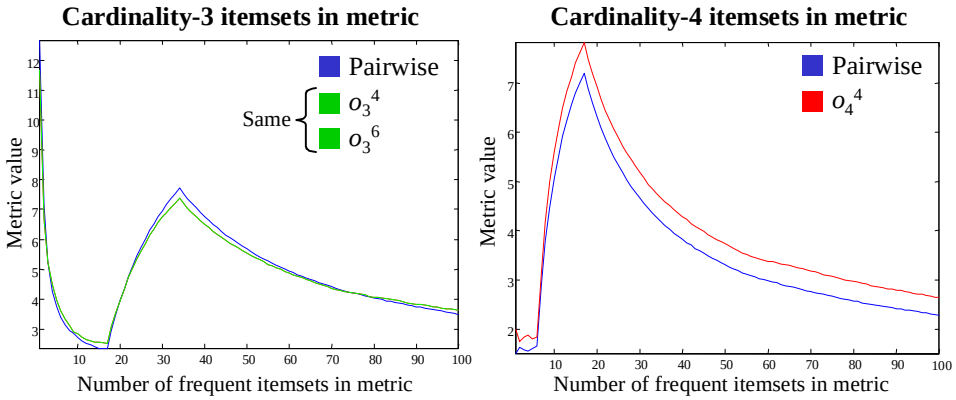


Figure 5-23: Minimum spanning tree itemset-degree metrics for "Genetic Algorithms and Neural Networks" data set.

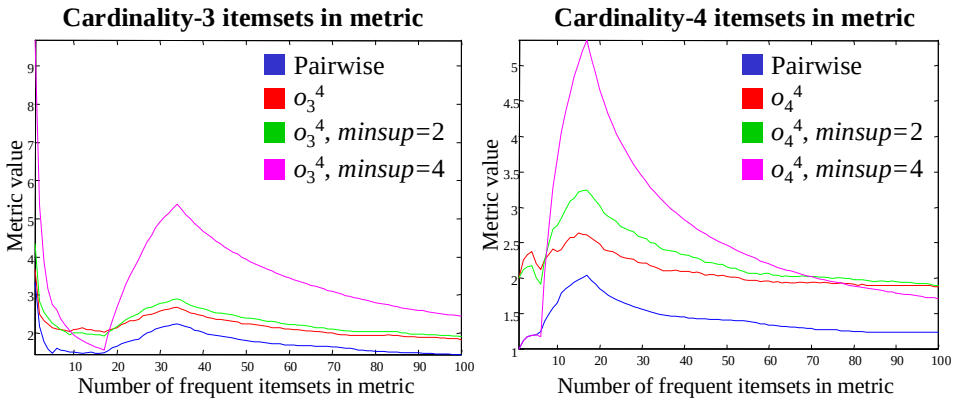


Figure 5-24: Minimum spanning tree itemset-degree metrics for "Quantum Gravity and Strings" data set.

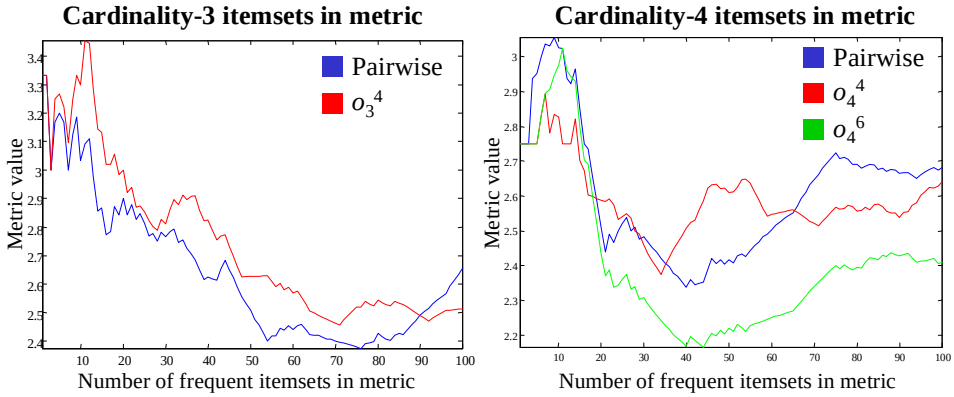


Figure 5-25: Minimum spanning tree itemset-degree metrics for "Quantum Gravity and Strings" data set, with bibliographic coupling.

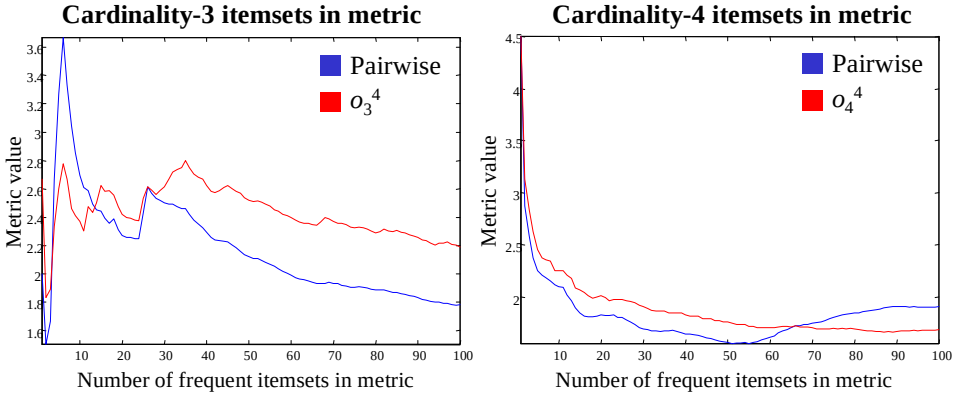


Figure 5-26: Minimum spanning tree itemset-degree metrics for "Wavelets (1-100)" data set.

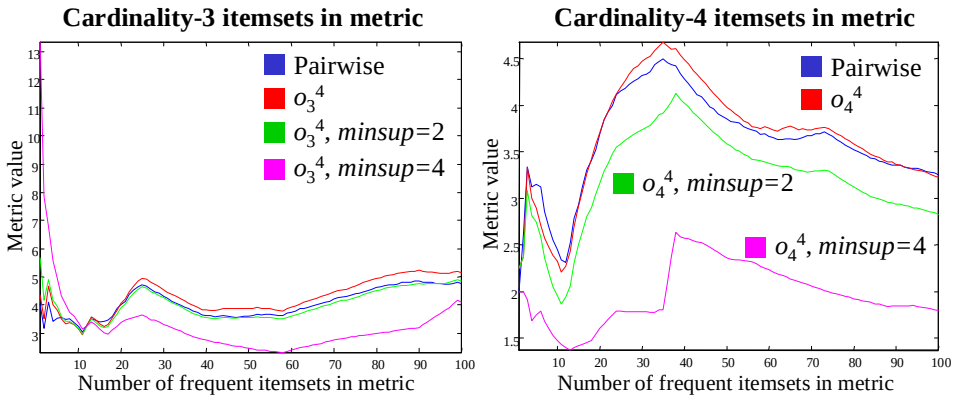


Figure 5-27: Minimum spanning tree itemset-degree metrics for "Wavelets (1-500)" data set.

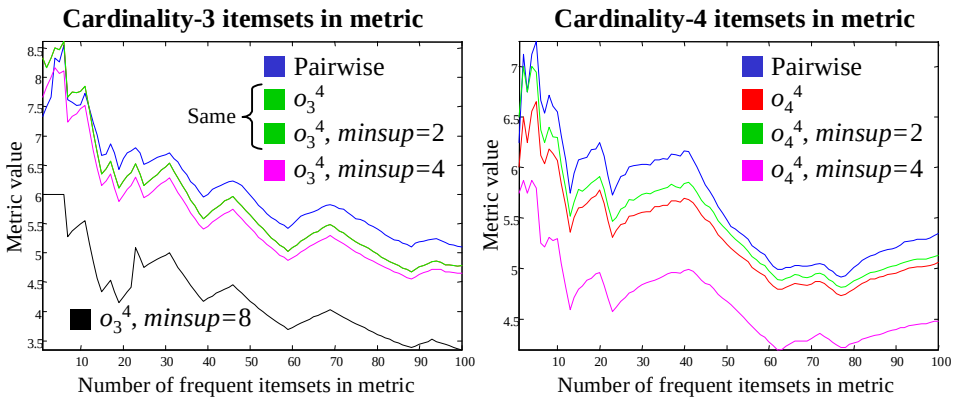


Figure 5-28: Minimum spanning tree itemset-degree metrics for "Wavelets and Brownian" data set.

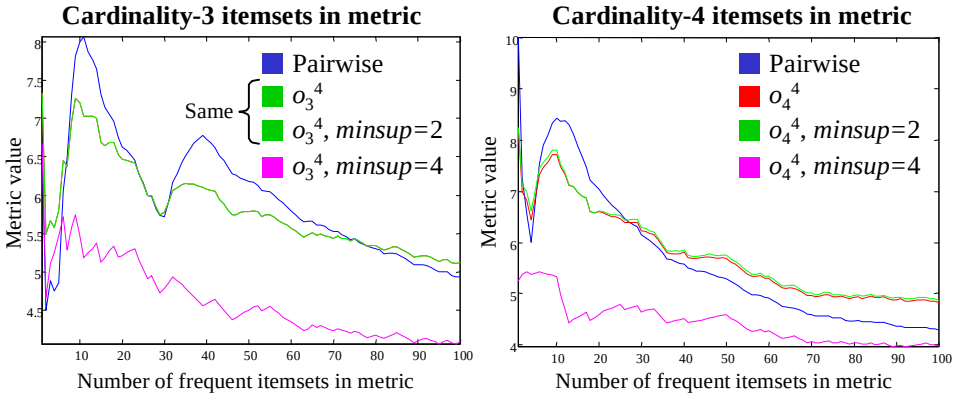


Figure 5-29: Minimum spanning tree itemset-degree metrics for "Wavelets and Brownian" data set, with bibliographic coupling.

Figures 5-30 through 5-39 show minimum spanning tree itemset-influence metric results for the SCI data sets. Each figure compares standard pairwise distances, to hybrid pairwise/higher-order distances. Standard distances are computed as in Eq. (5.5). Hybrid distances are computed as in Eq. (5.6), with itemset cardinality $\chi = 4$ and support nonlinearity $T(\zeta) = \zeta^4$. The influence metric applies the single most frequent itemset of cardinality $\chi = 4$.

The figures label the root chosen for the itemset-influence metric. Roots were chosen somewhat subjectively as the document with the most influence in the network, through a combination of high vertex degree and closeness to graph center. The particular choices of root raise no questions as to the validity of the resulting metric values.

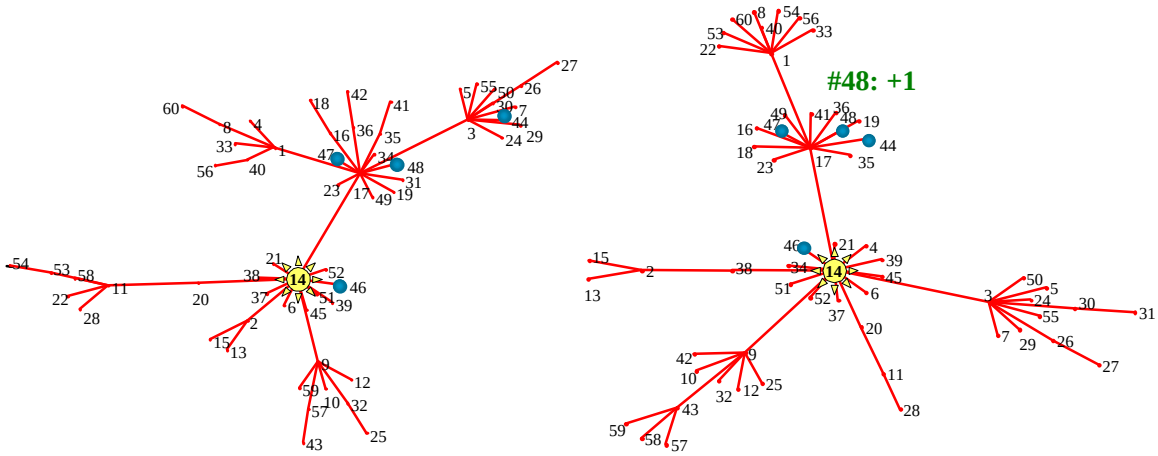
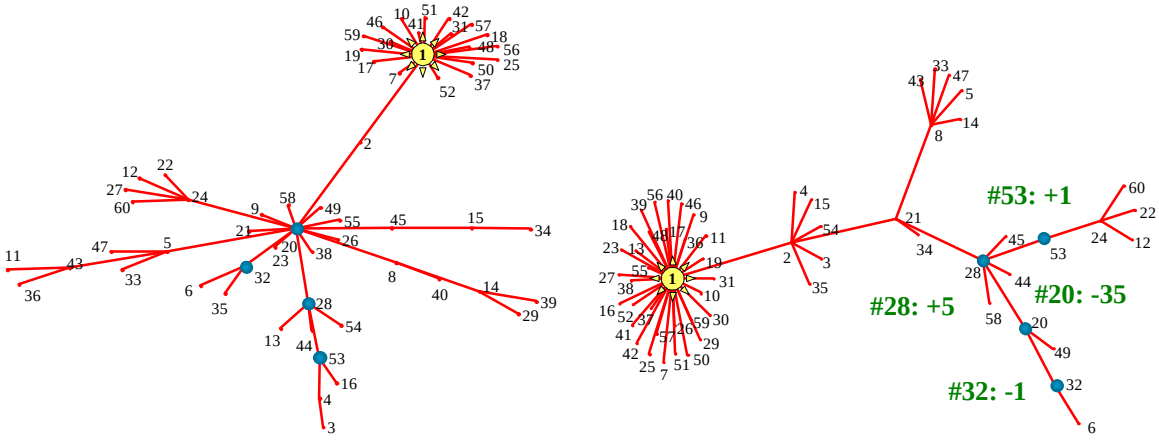
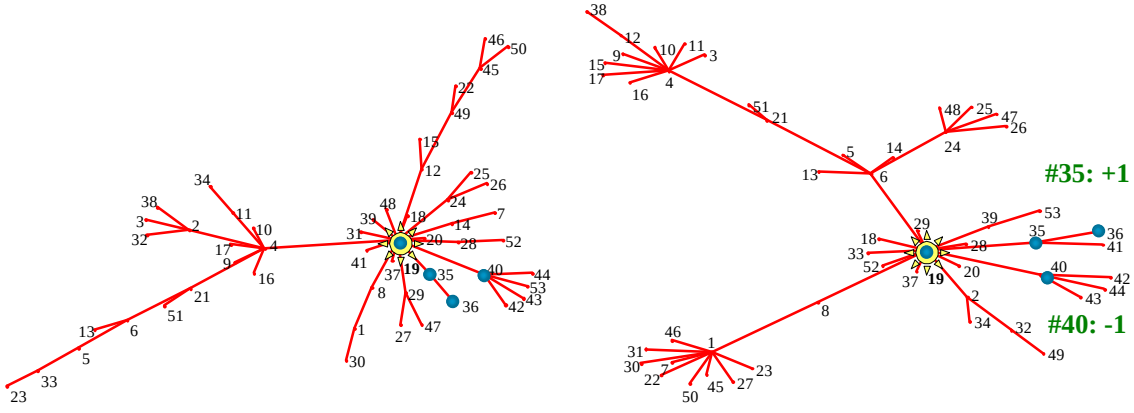


Figure 5-30: Standard versus hybrid distances in minimum spanning tree visualization with most frequent cardinality-4 itemset, for "Adaptive Optics" data set.



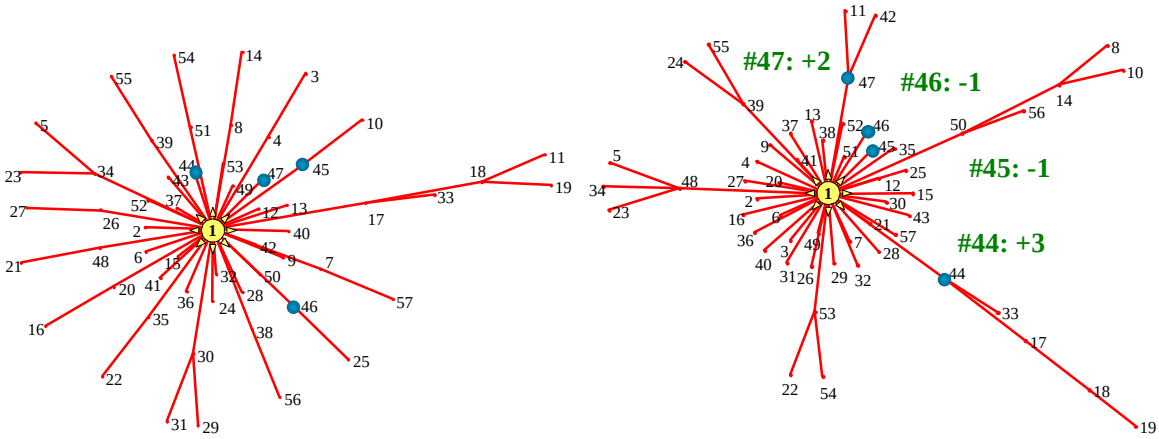
Pairwise distances o_4^4 **hybrid distances**

Figure 5-31: Standard versus hybrid distances in minimum spanning tree visualization with most frequent cardinality-4 itemset, for "Adaptive Optics" data set, with bibliographic coupling.



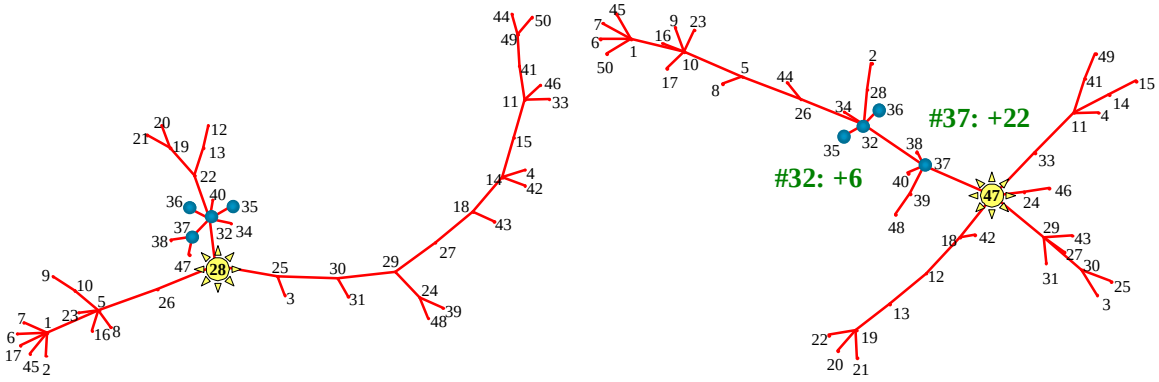
Pairwise distances o_4^4 **hybrid distances**

Figure 5-32: Standard versus hybrid distances in minimum spanning tree visualization with most frequent cardinality-4 itemset, for "Collagen" data set.



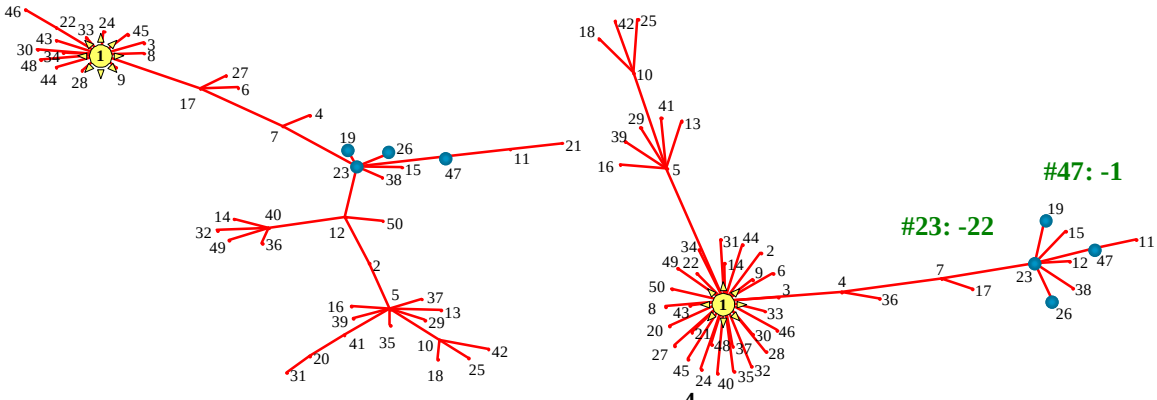
Pairwise distances o_4^4 **hybrid distances**

Figure 5-33: Standard versus hybrid distances in minimum spanning tree visualization with most frequent cardinality-4 itemset, for "Genetic Algorithms and Neural Networks" data set.



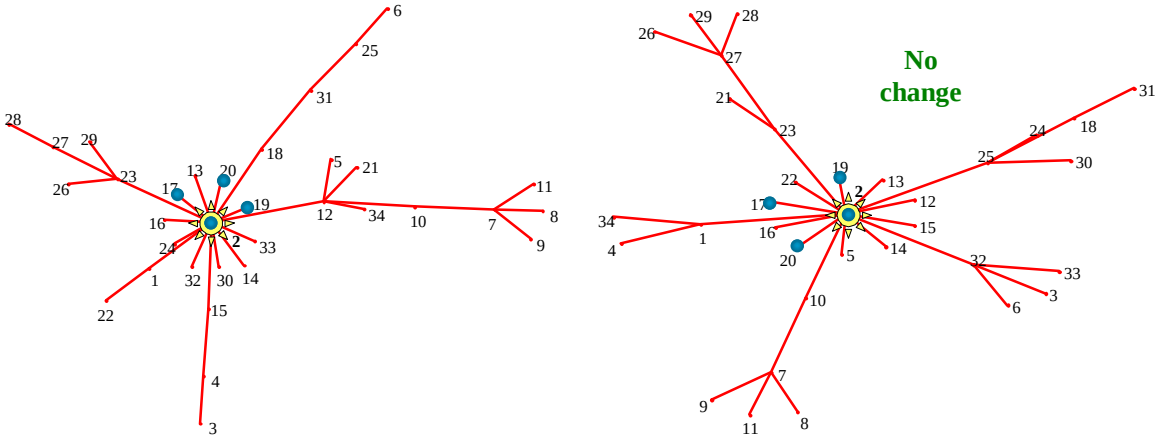
Pairwise distances o_4^4 **hybrid distances**

Figure 5-34: Standard versus hybrid distances in minimum spanning tree visualization with most frequent cardinality-4 itemset, for "Quantum Gravity and Strings" data set.



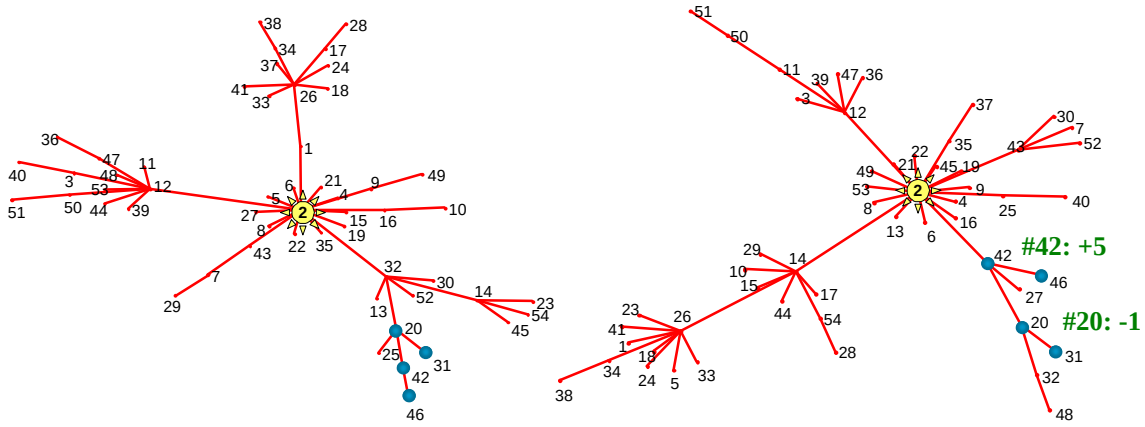
Pairwise distances o_4^4 **hybrid distances**

Figure 5-35: Standard versus hybrid distances in minimum spanning tree visualization with most frequent cardinality-4 itemset, for "Quantum Gravity and Strings" data set, with bibliographic coupling.



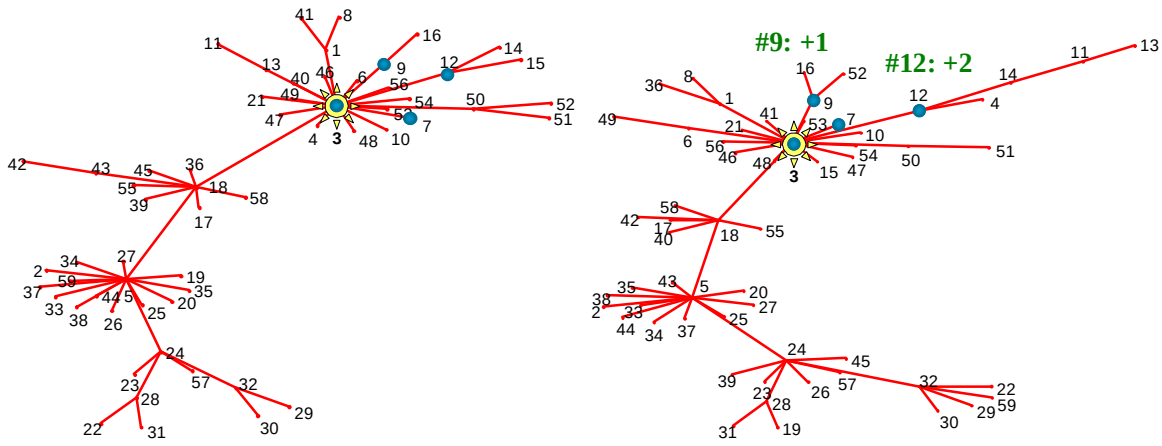
Pairwise distances o_4^4 **hybrid distances**

Figure 5-36: Standard versus hybrid distances in minimum spanning tree visualization with most frequent cardinality-4 itemset, for "Wavelets (1-100)" data set.



Pairwise distances o_4^4 **hybrid distances**

Figure 5-37: Standard versus hybrid distances in minimum spanning tree visualization with most frequent cardinality-4 itemset, for "Wavelets (1-500)" data set.



Pairwise distances o_4^4 **hybrid distances**

Figure 5-38: Standard versus hybrid distances in minimum spanning tree visualization with most frequent cardinality-4 itemset, for "Wavelets and Brownian" data set.

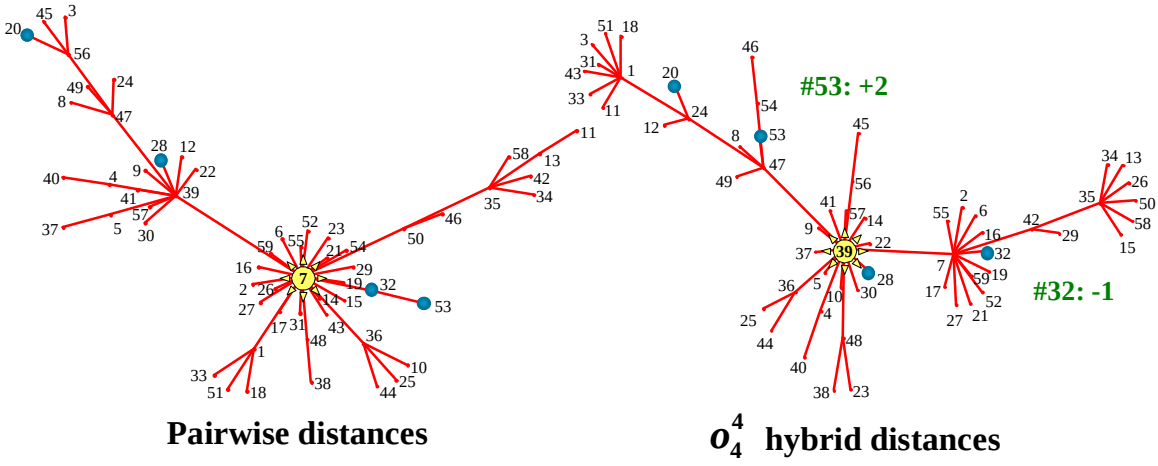


Figure 5-39: Standard versus hybrid distances in minimum spanning tree visualization with most frequent cardinality-4 itemset, for "Wavelets and Brownian" data set, with bibliographic coupling.

Table 5-5 summarizes the results of Figures 5-30 through 5-39. The itemset-influence metrics vary considerably among the test cases, being different between standard and hybrid distances for 9 out of 10 cases. For 3 of the cases, there is an itemset member with a distance-dependent change in metric value of roughly half the cardinality of the entire tree, reflecting a major restructuring of influence for that particular document.

For 6 out of 10 test cases, the net metric value over all itemset members is larger for hybrid distances. The net metric value is smaller for hybrid distances in only 2 out of the 10 test cases. The remaining 2 cases have either no net change or no individual changes.

Table 5-5: Comparison of minimum spanning tree itemset-influence metric for pairwise (P.W.) and hybrid (H.O.) distances.

Data set	P.W./H.O. deltas	Total delta
1	+1	+1
2	+5, +1, -1, -35	-30
3	+1, -1	0
4	+3, +2, -1, -1	+3
5	+22, +6	+28
6	-1,-22	-23
7	None	0
8	+5, -1	+4
9	+2, +1	+3
10	+2, -1	+1

These results suggest that there is somewhat of a tendency for hybrid pairwise/higher-order distances to increase the influence of frequent itemset members in the minimum spanning tree network visualization. But there can be significant exceptions to this tendency.

This section has applied the 3 new minimum spanning tree metrics proposed in Section 5.3 to various data sets extracted from the SCI citation database. In the next section, I introduce a new visualization of the minimum spanning tree. It visualizes multiple-resolution density estimates of the minimum spanning tree vertices, computed via the wavelet transform.

5.5 Landscape Visualization for Minimum Spanning Tree

Wise *et al* have proposed a novel visualization of document collections [Wise95]. They compute inter-document distances via text analysis, then project document points to the plane. A density is then computed for the resulting point scatterplot. The density surface is reminiscent of a landscape, in which high concentrations of documents form peaks, and low concentrations form valleys. Because peaks generally represent groups of documents with a common textual theme, the visualization is termed a "themescape."

In this section, I extend the themescape in visualizing the minimum spanning tree computed from higher-order co-citations. It is extended from a single density surface to a range of possible surfaces, corresponding to density estimates at various resolutions [Noel97]. The multiple-resolution density estimates are computed with the wavelet transform. The resulting visualization is interpreted as a spatial, hierarchical, fuzzy clustering of the minimum spanning tree vertices.

The themescape document landscape allows direct visualization of the density structure of a collection, rather than having to infer it from the individual points. This provides convenient analysis for information retrieval. While the document landscape visualization has been applied to text analysis, I propose for the first time its application to citation analysis.

I include additional extensions of the document landscape that support information retrieval tasks. One is to embed the minimum spanning tree directly into the density surface, to allow interaction with documents in the tree. Another is to augment the visualization with graphical glyphs for members of frequent itemsets, to identify these

significant associations among documents. A third is to add bibliographic text for documents in the tree.

Consider the minimum spanning tree $MST(G) = (V, E_t)$ of distance graph G, with vertices V and tree edges $E_t = (u_t, v_t)$. The result of the force-directed vertex placement algorithm described in Section 5.2 is a set of spatial coordinates for V. In particular, the ordered pair (x_i, y_i) gives the coordinates of vertex $V_i \in V$, for spatial variables x and y.

For the document landscape visualization, the real-valued coordinates (x_i, y_i) are first normalized, via the linear transformations

$$\hat{x}_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{5.14}$$

and

$$\hat{y}_i = \frac{y_i - y_{min}}{y_{max} - y_{min}}. \tag{5.15}$$

Here $x_{min} \equiv \min(x_i) - b$ and $y_{min} \equiv \min(y_i) - b$, where b defines a border area around the minimum spanning tree.

The normalized coordinates $(\hat{x}_i, \hat{y}_i)$ are converted from real to integer values via

$$\tilde{x}_i = \lfloor (n-1)\hat{x}_i + 0.5 \rfloor + 1 \tag{5.16}$$

and

$$\tilde{y}_i = \lfloor (n-1)\hat{y}_i + 0.5 \rfloor + 1. \tag{5.17}$$

Here $\tilde{x}_i \in [1, n]$ and $\tilde{y}_i \in [1, n]$ are discrete coordinates, where n is the length of a data vector corresponding to the ranges $x_i \in [x_{min} - b, x_{max} + b]$ and $y_i \in [y_{min} - b, y_{max} + b]$,

respectively. Eqs. (5.16) and (5.17) potentially loose information, via the truncations, though the upper bound $\left[\sqrt{2}(n-1)\right]^{-1}$ for the loss decreases for increasing n.

The discrete coordinates define an $n \times n$ binary coordinate matrix $\mathbf{C}$, as

$$c_{j,k} = \begin{cases} 1 & \tilde{x}_i = j \text{ and } \tilde{y}_i = k \\ 0 & \text{otherwise.} \end{cases} \tag{5.18}$$

In the language of signal processing, the binary coordinate matrix $\mathbf{C}$ results from pulse code modulation of the original real-valued coordinates, which themselves imply an impulsive function of the continuous spatial variables x and y.

From a signal processing perspective, the document landscape is a low-pass spatial filtering, or "blurring" of the coordinate matrix $\mathbf{C}$. The classical approach to filtering is based on the Fourier transform. However, it is generally accepted that the wavelet transform offers a much better framework for nonstationary signals. Because of the nonstationary nature of document landscapes, a wavelet approach is therefore appropriate.

The wavelet transform is a generalization of the Fourier transform in which the basis functions of the transformation are localized in both time and frequency [Daub92][Stra96]. In essence it is a signal representation which is a compromise between time resolution and frequency resolution. The wavelet representation is particularly useful for signals with non-smooth features. Wavelets provide a signal representation over multiple resolutions, decomposing the signal at course to fine scales.

There are 2 types of wavelets: scaling functions (father wavelets) and wavelet functions (mother wavelets). Scaling functions $\phi(x)$ have the property

$$\int \phi(x)dx = 1. \tag{5.19}$$

They define discrete low-pass filters with filter coefficients l_k given by

$$l_k = \frac{1}{\sqrt{2}} \int \phi(x) \phi(2x - k) dx . \tag{5.20}$$

They also obey the dilation equation

$$\phi(x) = \sqrt{2} \sum_k l_k \phi(2x - k) . \tag{5.21}$$

Wavelet functions $\psi(x)$ have the property

$$\int \psi(x) dx = 0 . \tag{5.22}$$

They define discrete high-pass filters with filter coefficients h_k given by

$$h_k = \frac{1}{\sqrt{2}} \int \psi(x) \phi(2x - k) dx . \tag{5.23}$$

They also obey the dilation equation

$$\psi(x) = \sqrt{2} \sum_k h_k \phi(2x - k) . \tag{5.24}$$

The low and high pass filter coefficients are related by the quadrature mirror relationship

$$h_k = (-1)^k l_{1-k} . \tag{5.25}$$

These filter coefficients are the impulse responses of the low and high pass filters. Explicit formulas for wavelet or scaling functions generally do not exist. Rather, the functions are defined in terms of limits of iterations of the scaling and wavelet dilation equations.

Figure 5-40 shows 3 important sets of scaling and wavelet functions – the Haar wavelet, the Daubechies 4th-order "d4" wavelet, and the nearly symmetric Daubechies 8th-order "s8" wavelet. The Haar wavelet was invented in 1910, and is the first known

example of a wavelet. It is the only orthogonal wavelet with compact support that is symmetric. However, it is not continuous. The d4 wavelet was the first continuous orthogonal wavelet with compact support. The s8 wavelet was constructed to be as symmetric as possible.

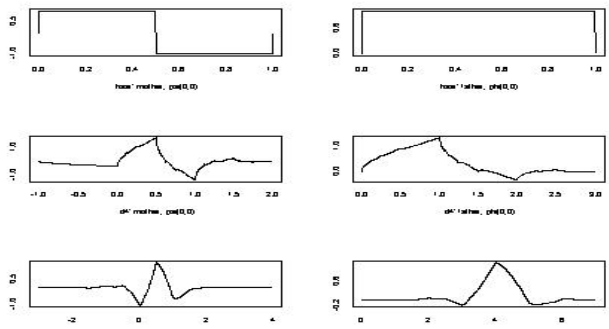


Figure 5-40: Haar, Daubechies, and nearly-symmetric Daubechies wavelets.

In the wavelet transform, the same basic wavelet shape is used throughout the decomposition. The transform is done rather with translated and scaled versions of the basic wavelet. A wavelet $\psi(x)$ is translated from origin by an amount b with $\psi(x-b)$. The translation is to the right for $b>0$, or to the left for $b<0$. Figure 5-41 shows the s8 wavelet with a translation $b=2$. A wavelet is then scaled in the x direction by an amount $a>0$ with $\psi(x/a)$. The wavelet is dilated (stretched) for $a>1$, or contracted (compressed) for $a<1$. Figure 5-41 also shows the s8 wavelet with a scaling $a=2$.

The orthogonal wavelet series approximation to a continuous signal $f(x)$, at scale 2^J (resolution 2^{-J}) is

$$f(x) \approx S_J(x) + D_J(x) + D_{J-1}(x) + \ldots + D_1(x). \qquad (5.26)$$

The orthogonal signal components $S_J(x), D_J(x), D_{J-1}(x), \ldots, D_1(x)$ are

$$\begin{aligned} S_J(x) &= \sum_k s_{J,k}\,\phi_{J,k}(x) \\ D_j(x) &= \sum_k d_{j,k}\,\psi_{j,k}(x) \end{aligned} \tag{5.27}$$

where k indexes the wavelet translations. Here $s_{J,k}, d_{J,k}, \ldots, d_{1,k}$ are the wavelet transform coefficients, and $\phi_{J,k}(x)$ and $\psi_{j,k}(x)$ are translated and scaled versions of the wavelet and scaling functions.

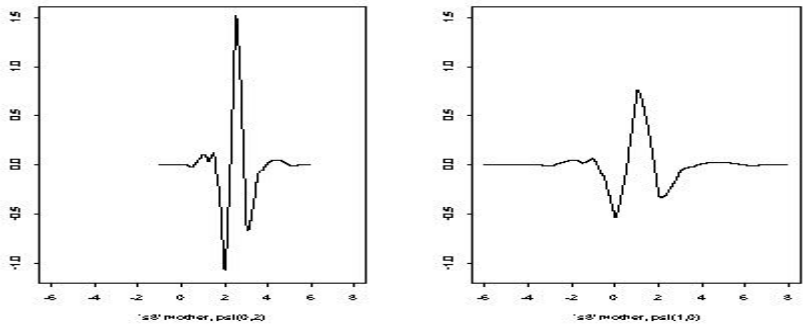


Figure 5-41: Translated and scaled wavelets.

We are generally interested in dyadic (power of 2) translations and scales, that is

$$\begin{aligned} \phi_{j,k}(x) &= 2^{-j/2}\,\phi\left(\frac{x - 2^j k}{2^j}\right) \\ \psi_{j,k}(x) &= 2^{-j/2}\,\psi\left(\frac{x - 2^j k}{2^j}\right) \end{aligned} \tag{5.28}$$

Here the wavelet coefficients $s_{J,k}$ and $d_{j,k}$ are given by

$$\begin{aligned} s_{J,k} &\approx \int \phi_{J,k}(x) f(x)\,dx \\ d_{j,k} &\approx \int \psi_{j,k}(x) f(x)\,dx,\ j = 1,2,\ldots,J \end{aligned} \tag{5.29}$$

The discrete wavelet transform computes the coefficients of the wavelet series approximation for a discrete signal $\mathbf{f} = \text{transpose}(f_1, f_2, \ldots, f_n)$. The transform is mathematically equivalent to the multiplication of $\mathbf{f}$ by an orthogonal matrix $\mathbf{W}$:

$$\mathbf{w} = \mathbf{W}\mathbf{f}\,, \tag{5.30}$$

where $\mathbf{w} = \text{transpose}(w_1, w_2, \ldots, w_n)$ are the wavelet coefficients. Likewise, the inverse transform is equivalent to

$$\mathbf{f} = \mathbf{W}^{-1}\mathbf{w}\,. \tag{5.31}$$

However, an algorithm exists for the wavelet transform that is faster than matrix multiplication, known as the pyramid algorithm [Mall89]. It involves the direct application of low-pass and high-pass filters corresponding to the wavelet and scaling functions. It also employs downsampling (decimation by 2) of the filter outputs, i.e. the removal of every other sample point. The pyramid algorithm is shown in Figure 5-42. It has algorithmic complexity $O(n)$. The linearity constant is on the order of the logarithm of the number of filter coefficients, and is typically quite small.

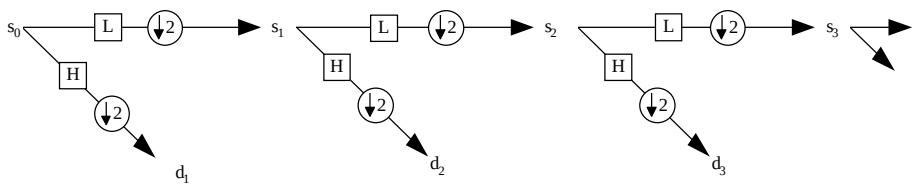


Figure 5-42: Fast pyramid algorithm for computing the discrete wavelet transform.

The input to the pyramid algorithm $\mathbf{s}_0$ is the values of the discrete signal $s_{0,i} = f_i$. The output is the detail wavelet coefficients $\mathbf{d}_j = \text{transpose}(d_{j,1}, d_{j,2}, \ldots, d_{j,n/2^j})$ and the

smooth wavelet coefficients $\mathbf{s}_J = \text{transpose}\left(s_{J,1}, s_{J,2}, \ldots, s_{J,n/2^J}\right)$. The inverse wavelet transform is simply the reverse of the pyramid algorithm. Downsampling is replaced by upsampling, that is, the padding of every other sample point with zero.

Figure 5-43 shows wavelet approximations to a signal at various resolutions, as described by Eq. (5.26). Here the input signal *f(x)* represents a row or column of the minimum spanning tree $n \times n$ binary coordinate matrix **C** given in Eq. (5.18). Recall that each signal impulse for this matrix corresponds to a tree vertex, where the original continuous function of 2 spatial variables has been pulse-code modulated. The interpretation from the figure is that a wavelet signal approximations at a given resolution provides a minimum spanning tree vertex density estimations at that resolution.

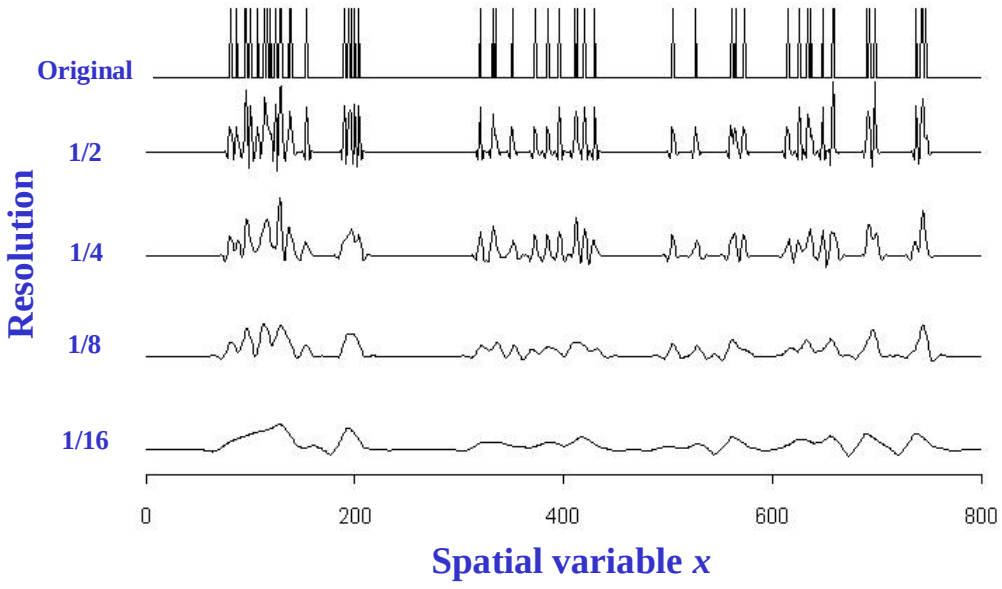


Figure 5-43: Wavelet approximation to signal at various resolutions.

The minimum spanning tree vertices have coordinates in 2 spatial dimensions, so that in the language of signal processing they form an image. Wavelets must therefore be extended to 2 dimensions, to enable an image transform. The basis functions for the 2-dimensional wavelet extension are tensor products of the respective one-dimensional basis functions. In particular, the tensor products of one-dimensional father and mother wavelets in the horizontal and vertical directions result in 4 different types of 2-dimensional wavelets:

$$\begin{aligned}\Phi(x,y) &= \phi_h(x)\phi_v(y)\\ \Psi^v(x,y) &= \psi_h(x)\phi_v(y)\\ \Psi^h(x,y) &= \phi_h(x)\psi_v(y)\\ \Psi^d(x,y) &= \psi_h(x)\psi_v(y).\end{aligned} \tag{5.32}$$

One-dimensional father and mother wavelets are good at representing smooth and detail signal components, respectively. Thus $\Phi(x,y)$ is good at representing smooth parts of an image, $\Psi^v(x,y)$ is good at representing vertical image detail, $\Psi^h(x,y)$ is good at representing horizontal image detail, and $\Psi^d(x,y)$ is good at representing diagonal image detail. Figure 5-44 shows $\Phi(x,y)$, $\Psi^v(x,y)$, $\Psi^h(x,y)$, and $\Psi^d(x,y)$ for the Haar wavelet.

A continuous 2-dimensional function $F(x,y)$ can be written in terms of translated and scaled wavelet bases as

$$\begin{aligned}F(x,y) \approx &\sum_{m,n} s_{J,m,n}\Phi_{J,m,m}(x,y) + \sum_{j=1}^{J}\sum_{m,n} d^v_{j,m,n}\Psi^v_{j,m,n}(x,y)\\ &+ \sum_{j=1}^{J}\sum_{m,n} d^h_{j,m,n}\Psi^h_{j,m,n}(x,y) + \sum_{j=1}^{J}\sum_{m,n} d^d_{j,m,n}\Psi^d_{j,m,n}(x,y),\end{aligned} \tag{5.33}$$

where

$$\begin{aligned}
\Phi_{J,m,m}(x,y) &= 2^{-J}\Phi\left(2^{-J}x-m,2^{-J}y-n\right)\\
\Psi^{v}_{j,m,n}(x,y) &= 2^{-J}\Psi^{v}\left(2^{-J}x-m,2^{-J}y-n\right)\\
\Psi^{h}_{j,m,n}(x,y) &= 2^{-J}\Psi^{h}\left(2^{-J}x-m,2^{-J}y-n\right)\\
\Psi^{d}_{j,m,n}(x,y) &= 2^{-J}\Psi^{d}\left(2^{-J}x-m,2^{-J}y-n\right)
\end{aligned} \tag{5.34}$$

are the 2-dimensional basis functions with dyadic translation and scaling. The 2-dimentional wavelet transform coefficients are given by

$$\begin{aligned}
s_{J,m,n} &\approx \iint \Phi_{J,m,m}(x,y)F(x,y)dxdy\\
d^{v}_{j,m,n} &\approx \iint \Psi^{v}_{j,m,n}(x,y)F(x,y)dxdy\\
d^{h}_{j,m,n} &\approx \iint \Psi^{h}_{j,m,n}(x,y)F(x,y)dxdy\\
d^{d}_{j,m,n} &\approx \iint \Psi^{d}_{j,m,n}(x,y)F(x,y)dxdy.
\end{aligned} \tag{5.35}$$

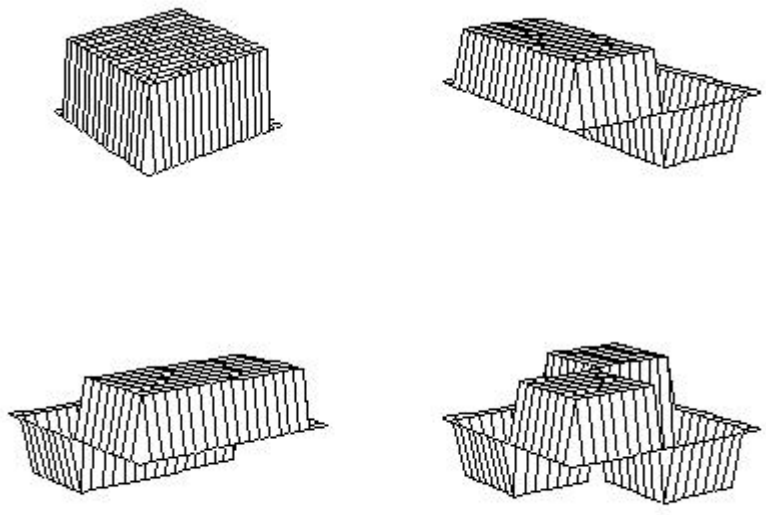


Figure 5-44: One smooth (upper left) and 3 detail basis functions for 2-dimensional extension of Haar wavelet.

A version of the pyramid algorithm exists for discrete 2-dimensional signals, i.e. images [Mall89]. Figure 5-45 shows one stage of this algorithm. The input image $\mathbf{s}_0$ is filtered along its rows with the one-dimensional low-pass and high-pass filters, and the 2 outputs are downsampled along their columns.

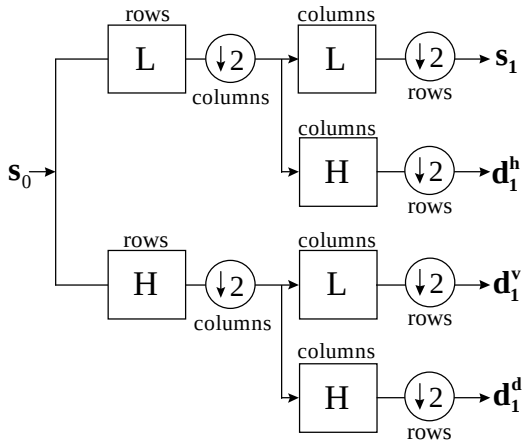


Figure 5-45: First stage of 2-dimensional pyramid algorithm.

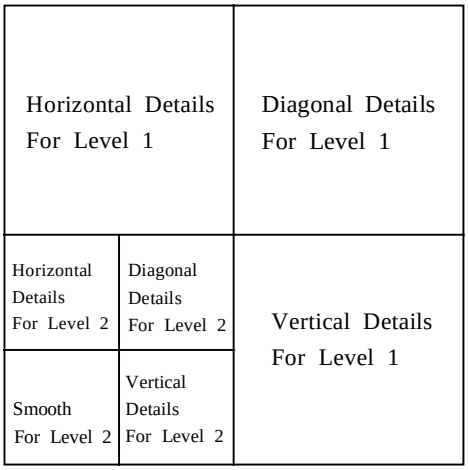


Figure 5-46: Organization of wavelet coefficients for 2 levels of the 2-dimensional transform.

The initial 2 filter outputs are then each filtered along their columns with the one-dimensional low-pass and high-pass filters, and the 4 outputs are downsampled along their rows. The result is the smooth and horizontal, vertical, and diagonal detail images for the next lower resolution level. Because of the downsampling at each stage, the 2-dimensional wavelet coefficients can be arranged as shown in Figure 5-46.

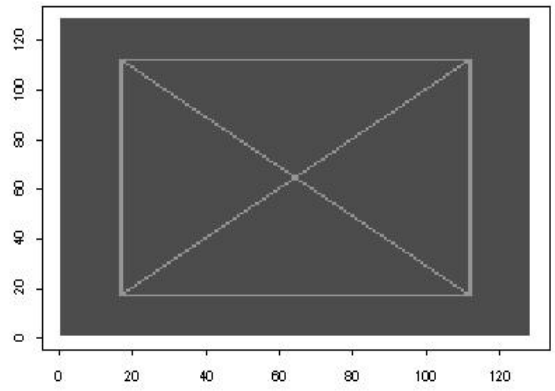


Figure 5-47: Image of box with diagonals.

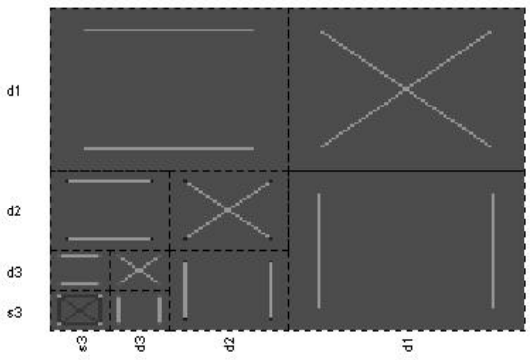


Figure 5-48: 3-Level wavelet transform of box image.

Figure 5-47 shows an image of a box with diagonals. Figure 5-48 is a 3-level wavelet transform of the box image. The transform clearly shows the horizontal, vertical, and diagonal details at each level of resolution and the smoothed image at the 3rd level.

Figure 5-49 shows wavelet-based minimum spanning tree density estimates at various resolutions. In terms of image processing, each density estimate is a low-pass spatially filtered ("smeared") version of the initial minimum spanning tree image. Successive applications of the wavelet low-pass filter causes nearby points to merge into single smeared peaks. These merges can be interpreted as a type of spatial, fuzzy clustering. Lower resolution peaks cover nearly the same areas as multiple peaks at higher resolutions. This is analogous to hierarchical clusters.

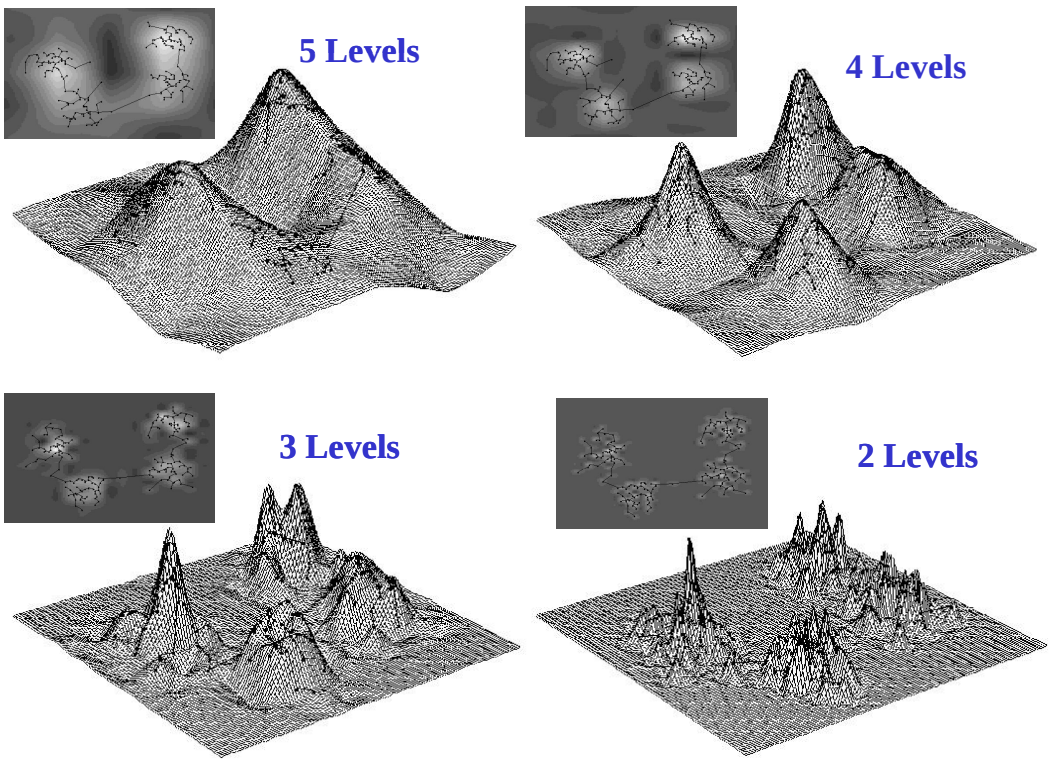


Figure 5-49: Outputs of various levels of wavelet low-pass spatial filter for minimum spanning tree density visualization.

For a wavelet-generated document landscape at a given resolution, the minimum spanning tree vertices can be embedded in the landscape surface. This allows visualization of the fuzzy spatial clusters in the minimum spanning tree, along with crisp vertices for navigation in information retrieval. The embedding of the tree allows a direct 3-dimensional interaction with the document points, while the landscape surface provides depth cues that alleviate the disorientation typical of visualizing points in 3-dimensions. The document landscape visualization with embedded minimum spanning tree is shown in Figure 5-50.

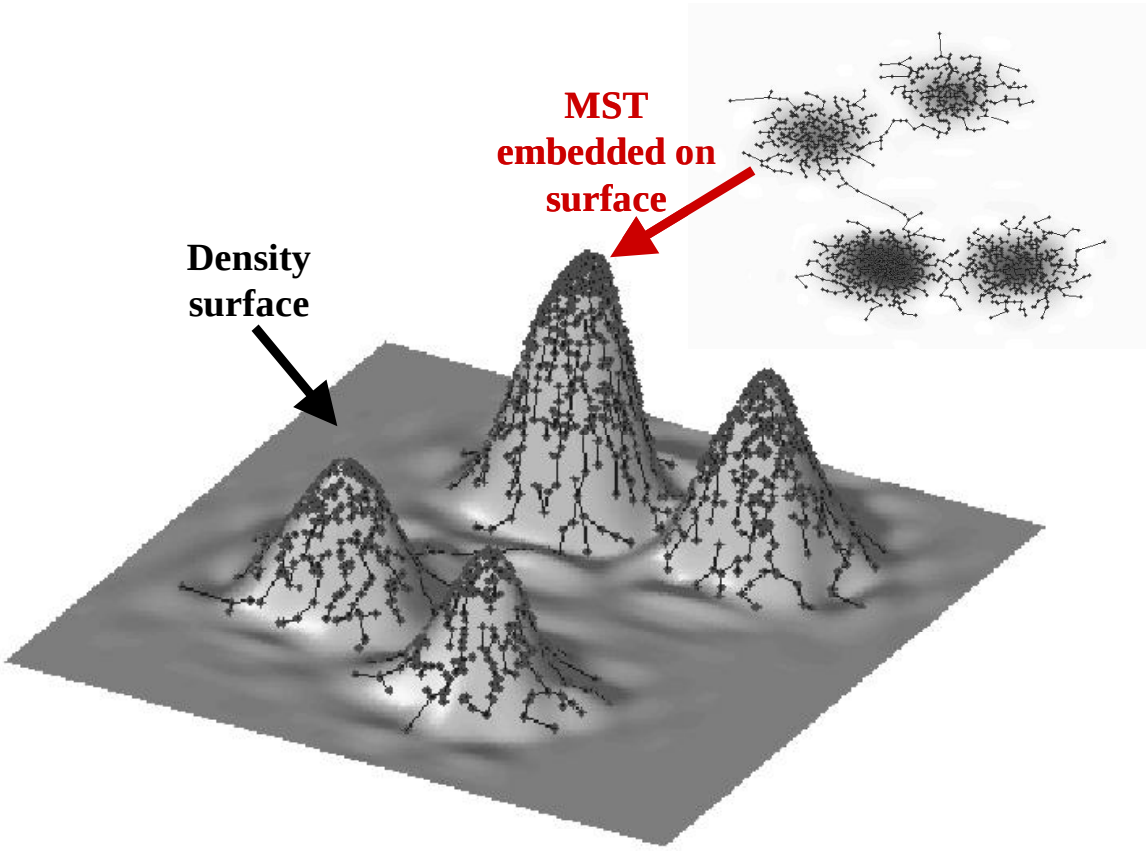


Figure 5-50: Minimum spanning tree embedded on its density landscape surface.

Although it may not lead to competitive clustering algorithms, there is an interesting interpretation of the landscape density surface as a clustering mechanism in the traditional sense. This interpretation is illustrated in Figure 5-51. Document points could serve as initial positions for optimization algorithms, so that the landscape local minima become optimization basins of attraction. For a deterministic optimization algorithm (e.g. hill climbing), the cluster basin boundaries would be crisp, while for a probabilistic algorithm (e.g. simulated annealing), they would be fuzzy. Cluster boundaries could also serve as decision surfaces for document classification.

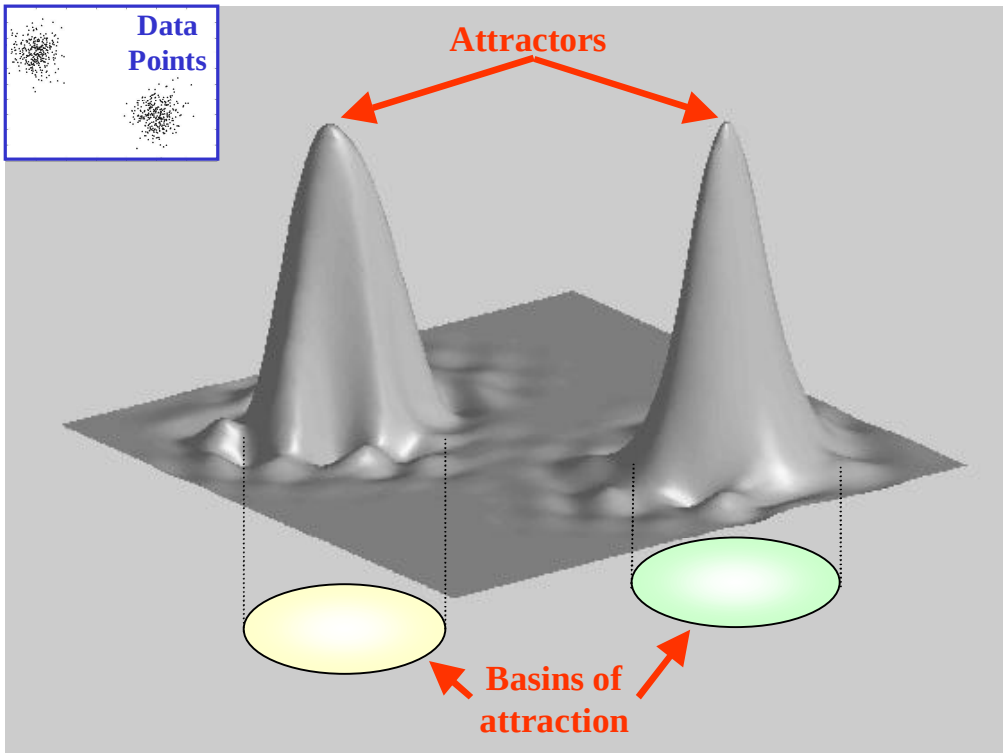


Figure 5-51: Interpretation of landscape surface as true clustering mechanism via local attractors of optimization algorithms.

There is a more direct way of generating crisp clusters from the fuzzy spatial document landscape, as shown in Figure 5-52. Here I apply a threshold to the grayscale landscape image, so that the resulting binary image shows which portions of the landscape surface lie above the threshold. Connected components of the binary image correspond to landscape peaks, which in turn correspond to individual document clusters. As before, the resolution of the landscape corresponds to the level in the cluster hierarchy.

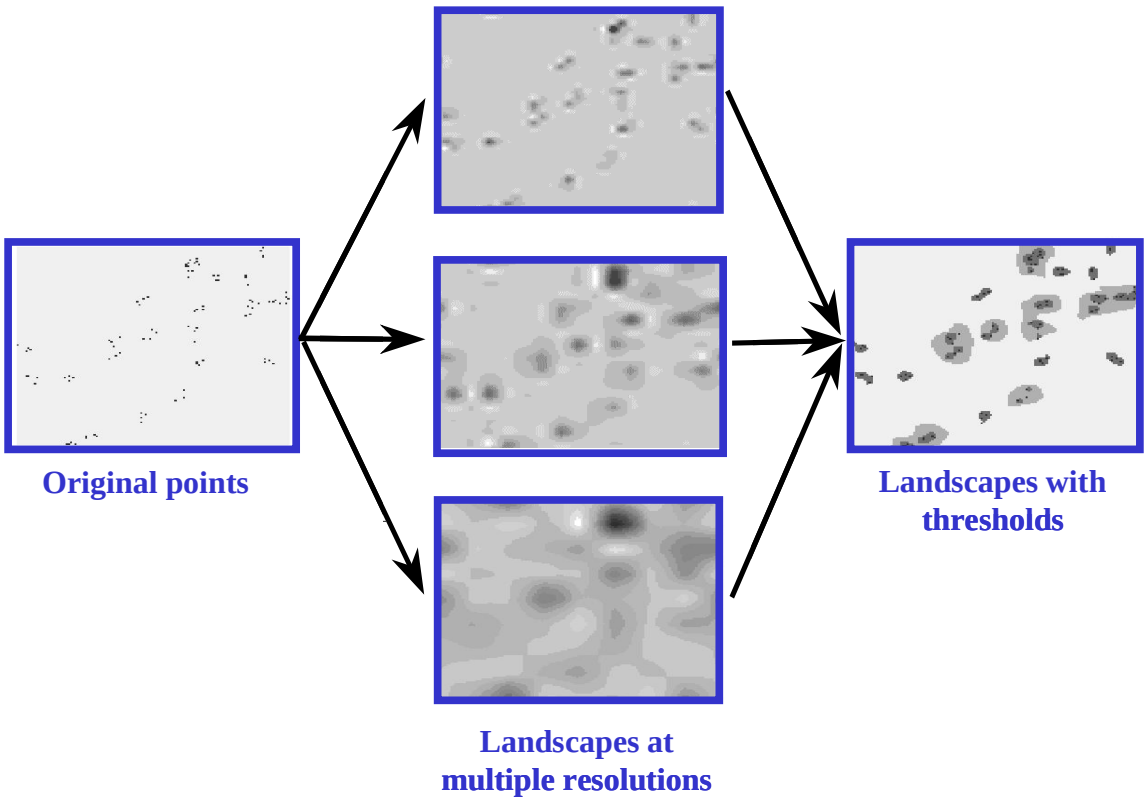


Figure 5-52: Application of spatial thresholds to document landscape to generate crisp clusters.

Interestingly, this image thresholding approach to clustering is similar to an even more straightforward clustering method for the minimum spanning tree vertices. In this

more straightforward method, a distance threshold is applied to the minimum spanning tree edges. The application of such a threshold is usually equivalent to single-linkage clustering. But I propose the application of the threshold not to the original edge distances, but to the edge distances induced by the force-directed layout algorithm described in Section 5.2.

In the minimum spanning tree layout, highly influential documents are generally spaced further apart, because of the greater net repulsive force of the many documents linked to them in the tree. Thus edges between highly influential documents are more likely to be removed by the application of the threshold. But edges between these central, highly influential documents generally represent relatively small co-citation based distances.

This is in contrast to more typical approaches to clustering, in which the highly influential documents would tend to be together in clusters, because of the relatively small distances between them. Thus my proposed approach is a novel type of clustering in which clusters are oriented to highly influential documents, and the highly influential documents themselves are placed in separate clusters.

This section concludes Chapter 5. In this chapter, I have applied hybrid pairwise/higher-order co-citation distances to minimum spanning tree document visualizations. I also proposed a number of extensions to the minimum spanning tree visualization, including a novel wavelet-based multiple-resolution landscape visualization of tree vertex density.

Chapter 6

Summary, Conclusions and Future Work

In this chapter, I summarize my work and describe the conclusions that can be drawn from it. This appears in the next section. In section 6.2, I discuss ideas for extending this work in the future.

6.1 Summary and Conclusions

In this dissertation, I have proposed new methods for visualizing collections of hypertext documents, leading to enhanced understanding of relationships among documents that are returned by information retrieval systems. 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.

The distances I propose are computed from higher-order similarities, but still retain a pairwise structure. These pairwise/higher-order hybrid distances allow the direct application of standard visualization tools such as clustering dendrograms and minimum spanning trees. However, they require much less complex user interaction compared to that of working directly with all frequent higher-order itemsets. The hybrid distances are

computationally feasible via previously proposed fast algorithms for computing frequent itemsets.

I provide a theoretical guarantee that consistency between clusters and frequent itemsets is attainable under the new hybrid distances. The guarantee is that there is always a sufficient degree of nonlinearity one can apply to itemset supports such that a more frequent itemsets forms a cluster at the expense of a less frequent itemset that overlaps it. While the guarantee does not include an upper bound on the necessary degree of nonlinearity, empirical results show that it is generally fairly low.

A similar guarantee can be supplied for the minimum spanning tree, in that there is always a sufficient nonlinearity degree such that a more frequent itemset will not be disconnected in the tree by a less frequent one. I show that for typical distributions of itemset supports, frequencies of occurrence of larger supports are very small. This contributes to the ability of the new hybrid distances to produce clusters or minimum spanning trees consistent with frequent itemsets.

I also propose 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. I introduce new augmentations of the dendrogram to support the information retrieval process, by adding document-descriptive text and glyphs for members of frequent itemsets. I also propose a metric for measuring the extent to which a clustering is consistent with frequent itemsets, which is based on the cardinality of the smallest cluster that contains all itemset members.

The minimum spanning tree has previously been proposed for visualizing co-citation relationships, the tree being interpreted as a network of document influences. In my work, I show that the new hybrid pairwise/higher-order distances tend to make the minimum spanning tree more consistent with frequent itemsets.

That is, when the hybrid distances are applied, frequent itemsets are more likely to be connected within the resulting trees, as measured by a metric that I propose based on numbers of itemset connected components. There is a slight tendency for the number of direct influences of frequent itemsets to increase, per a metric I propose based on itemset vertex degree. There is also a somewhat unpredictable tendency for the overall influence of a frequent itemset member to increase within the minimum spanning tree network, based on the number of its descendents in the tree.

This work represents the first known 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.

3.2 Future Work

I now suggest ideas for extending this work. These ideas fall under the general headings of user-oriented clustering, the inference of association rules, the role of

maximal frequent itemsets, extensions of visualizations to higher dimensions, and the inference of citation semantics.

In *user-oriented clustering*, the user iteratively provides *a priori* domain knowledge to guide the clustering process. This is generally accomplished by the weighting of document pairs according to the importance of them being together in a cluster. The application of higher-order co-citation similarities allows sets of arbitrary cardinality to be weighted, providing a much richer form of cluster orientation.

For example, suppose documents *A*, *B*, and *C* should be clustered together, but *B*, *C*, and *D* should not. This orientation cannot be accomplished by mere pairwise weights. A high weight applied for the pair (*B*, *C*) on behalf of the triple (*A*, *B*, *C*) inadvertently increases the weight for (*B*, *C*, *D*). But with higher-order co-citations, such weighting of the triples is trivial.

In association mining, the computation of frequent itemsets is often the first step in computing *association rules*, in which the presence of one itemset implies with some strength the existence of another, for non-overlapping itemsets. My results suggest that perhaps association rules can be inferred from the clustering dendrogram with the application of hybrid pairwise/higher-order distances.

An important type of frequent itemset in association mining is known as a *maximal* frequent itemset. This is a frequent itemset that is a subset of no other frequent itemset. In terms of the itemset lattice with partial order imposed by the subset relation, maximal frequent itemsets not “less than” any other itemset in the lattice. The role of maximal frequent itemsets in clustering with hybrid pairwise/higher-order distances remains to be explored.

In the landscape visualization I proposed for the minimum spanning tree, the tree is embedded in a vertex density estimate computed with the wavelet transform. The intersection of the surface with a threshold plane yields contour lines in the plane. The contours enclose areas containing clusters of minimum spanning tree vertices.

Now consider the extension of this to an additional spatial dimension. First position the minimum spanning tree vertices in a 3-dimensional volume rather than a plane. Then compute the wavelet-based vertex density in 3 dimensions. The application of a threshold volume to the 3-dimensional density results in contour surfaces, as opposed to the previous contour lines. These contour surfaces enclose volumes, each volume containing a cluster of vertices (documents). This is shown in Figure 6-1.

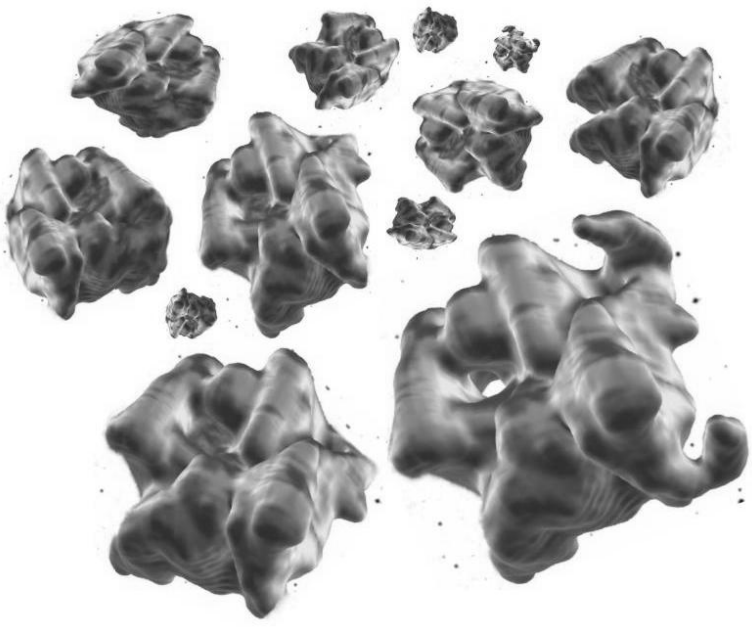


Figure 6-1: Extension of minimum spanning tree visualization to one higher spatial dimension.

Extending the minimum spanning tree landscape visualization should greatly improve the performance of algorithms for positioning tree vertices, since vertices would have more freedom in repositioning themselves during algorithm iterations. Perhaps the additional spatial dimension will also aid in the user's absorption of complex relationships in the minimum spanning tree visualization.

Currently the semantics of hypertext links such as science citations are lacking. The assumption is co-citations always imply some sort of similarity between documents, despite the common knowledge that the reasoning behind different citations or hyperlinks varies widely. Perhaps analyses of distributions of citations (links) within various documents, along with correlations between links and nearby text can help us make inferences about the meaning of the links.

Bibliography

[Agra93] R. Agrawal, T. Imielinski, A. Swami, "Mining Association Rules Between Sets of Items in Large Databases," in *Proceedings of the 1993 International Conference on the Management of Data*, eds. P. Buneman and S. Jajodia, Washington, DC, May 1993, pp. 207-216.

[Agra94] R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association Rules," in *Proceedings of the 20th International Conference on Very Large Databases*, eds. J. Bocca, M. Jarke and C. Zaniolo, Santiago, Chile, September 1994, pp. 487-499.

[Baez99] R. Baiza-Yates, B. Ribeiro-Neto, *Modern Information Retrieval*, ACM Press, New York, 1999.

[Bhuy91a] J. Bhuyan, V. Raghavan, "A Probablistic Retrieval Scheme for Cluster-Based Adaptive Information Retrieval," in *Proceedings of ML 91 – The 8th International Workshop on Machine Learning*, eds. L. Birnbaum and G. Collins, Evanston, IL, 1991, pp. 240-254.

[Bhuy91b] J. Bhuyan, V. Raghavan, J. Deogun, "Cluster-Based Adaptive Information Retrieval," in *Proceedings of the 24th Hawaii International Conference on System Sciences*, Volume I, eds. R. Sprague and B. Shriver, Kailua-Kona, HI, 1991, pp. 307-316.

[Bhuy97] J. Bhuyan, J. Deogun, V. Raghavan, "Algorithms for the Boundary Selection Problem," *Algorithmica*, 17, pp. 133-161, 1997.

[Page98] 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.

[Card99] S. Card, J. Mackinlay, B. Shneiderman (eds.), *Readings in Information Visualization: Using Visualization to Think*, Morgan Kaufmann, San Francisco, 1999.

[Chen99a] C. Chen, "Visualising Semantic Spaces and Author Co-Citation Networks in Digital Libraries," *Information Processing & Management*, 35, pp. 401-420, 1999.

[Chen99b] 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)*, chair J. Haake, Darmstadt, Germany, February, 1999, pp. 51-60.

[Corm96] T. Cormen, C. Leiserson, R. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1996.

[Daub92] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia, 1992.

[Eade84] P. Eades, "A Heuristic for Graph Drawing," *Congressus Numerantium*, 42, pp. 149-160, 1984.

[Fruc91] T. Fruchterman, E. Reingold, "Graph Drawing by Force-Directed Placement," *Software – Practice and Experience*, eds. D. Comer and A. Willings, 21, pp. 1129-1164, 1991.

[Garf78] E. Garfield, M. Malin, H. Small, "Citation Data as Science Indicators," in *Toward a Metric of Science: The Advent of Science Indicators*, eds. Y. Elkana, J.

Lederberg, R. Merten, A. Thackray, H. Zuckerman, John Wiley & Sons, New York, 1978, pp. 179-207.

[Garf79] E. Garfield, *Citation Index: Its Theory and Application in Science, Technology, and Humanities*, John Wiley & Sons, New York, 1979.

[Hafe99] A. Hafez, J. Deogun, V. Raghavan, "The Item-Set Tree: A Data Structure for Data Mining," in *Proceedings of Data Warehousing and Knowledge Discovery (DaWaK' 99)*, eds. M. Mohania and A. Tjoa, Florence, Italy, August 1999, pp. 183-192.

[Hart75] J. Hartigan, *Clustering Algorithms*, John Wiley & Sons, 1975.

[Hend95] R. Hendley, N. Drew, A. Wood, R. Beale, "Narcissus: Visualizing Information," in *Proceedings of Information Visualization ' 95 Symposium*, chairs N. Gershon and S. Eick, Atlanta, GA, October 1995, pp. 90-96.

[Henz00] M. Henzinger, "Link Analysis in Web Information Retrieval," *Bulletin of the Technical Committee on Data Engineering*, 23, special issue on Next-Generation Web Search, pp. 3-8, September 2000.

[Klei98] J. Kleinberg, "Authoritative Sources in a Hyperlinked Environment," in *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, chair H. Karloff, San Francisco, CA, January 1998, pp. 668-677.

[Lamp95] J. Lamping, R. Rao, P. Pirolli, "A Focus + Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies," in *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI ' 95)*, eds. I. Katz, R. Mack and L. Marks, Denver, CO, May 1995, pp. 401-408.

[Mack95] J. Mackinlay, R. Rao, S. Card, "An Organic User Interface for Searching Citation Links," in *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI ' 95)* eds. I. Katz, R. Mack and L. Marks, Denver, CO, May 1995, pp. 67-73.

[Mall89] S. Mallat, "A Theory For Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11, pp. 674-693, 1989.

[Mand83] B. Mandelbrot, *Fractal Geometry of Nature*, Freeman, New York, 1983.

[Mukh94a] S. Mukherjea, J. Foley, "Navigational View Builder: A Tool for Building Navigational Views of Information Spaces," in *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI ' 94)* eds. B. Adelson, S. Dumais and J. Olson, Boston, MA, April 1994, pp. 289-290.

[Mukh94b] S. Mukherjea, J. Foley, S. Hudson, "Interactive Clustering for Navigating in Hypermedia Systems," in *Proceedings of the ACM European Conference of Hypermedia Technology*, ed. M. Ley, Edinburgh, Scotland, September 1994, pp. 136-144.

[Munz95] T. Munzner, P. Burchard, "Visualizing the Structure of the World Wide Web in 3D Hyperbolic Space," in *Proceedings of the VRML ' 95 Symposium*, eds. D. Nadeau, J. Moreland, San Diego, CA, December 1995, pp. 33-38.

[Munz97] T. Munzner, "H3: Laying Out Large Directed Graphs in 3D Hyperbolic Space," in *Proceedings of the 1997 IEEE Symposium on Information Visualization*, chairs R. Moorhead and N. Johnston, Phoenix, AZ, October 1997, pp. 2-10.

[Noel97] S. Noel, H. Szu, "Multiple-Resolution Clustering for Recursive Divide and Conquer," in *Proceedings of Wavelet Applications IV*, ed. H. Szu, Orlando, FL, April 1997, pp. 266-279.

[OSTI00] Office of Scientific and Technical Information (OSTI), U. S. Department of Energy, http://www.osti.gov/ostipg.html, last accessed May 2000.

[Robe91] G. Robertson, S. Card, J. Mackinlay, "Cone Trees: Animated 3D Visualizations of Hierarchical Information," in *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI ' 91)* eds. S. Robertson, G. Olson and J. Olson, New Orleans, LA, 1991, pp. 189-194.

[Rysz98] R. Michalski, I. Bratko, M. Kubat (eds.), *Machine Learning and Data Mining*, John Wiley & Sons, New York, 1998.

[Smal73] 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.

[Smal93] H. Small, "Macro-Level Changes in the Structure of Co-Citation Clusters: 1983-1989," *Scientometrics*, 26, pp. 5-20, 1993.

[Stra96] G. Strang, T. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge, Wellesley, MA, 1996.

[Terv98] L. Terveen, W. Hill, "Finding and Visualizing Inter-Site Clan Graphs," in *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI ' 98)* chairs C.-M. Karat and A. Lund, Los Angeles, CA, April 1998, pp. 448-455.

[Tuft91] E. Tufte, *Envisioning Information*, Graphics Press, Milford, MA, 1991.

[VanR90] A. Van Raan, "Fractal Dimension of Co-Citations," *Nature*, 347, p. 626, 1990.

[Vena94] W. Venables, B. Ripley, *Modern Applied Statistics with S-Plus*, Springer-Verlag, Berlin, 1994.

[Whit89] H. White, K. McCain, "Bibliometrics," *Annual Review of Information Science and Technology*, 24, pp. 119-186, 1989.

[Whit98] H. White, K. McCain, "Visualizing a Discipline: An Author Co-Citation Analysis of Information Science, 1972-1995," *Journal of the American Society for Information Science*, 49, pp. 327-356, 1998.

[Wise95] 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," chairs N. Gershon and S. Eick, in *Proceedings of Information Visualization ' 95 Symposium*, Atlanta, 1995, pp. 51-58.

[WOS00] Institute for Scientific Information's Web of Science available at http://www.isinet.com/isi/products/citation/wos/, last accessed November 2000.

Appendix A

Clustering Metrics for Standard versus Hybrid Distances

This appendix gives detailed results for the experiments conducted in Chapter 3, Section 3.5. In particular, it gives clustering metrics for each test case, along with the corresponding choice of experimental inputs. The results are presented Tables A-1 through A-10, corresponding to the 10 data sets employed.

Table A-1: Clustering metrics for "Adaptive Optics" data set. Red and black text marks standard-vs-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	1.0000
complete	o_3^4	$o_3, k=1$	1.0000
complete	o_3^6	$o_3, k=1$	1.0000
average	pairwise	$o_3, k=1$	1.0000
average	o_3^4	$o_3, k=1$	1.0000
average	o_3^6	$o_3, k=1$	1.0000
single	pairwise	$o_3, k=1$	1.0000
single	o_3^4	$o_3, k=1$	1.0000
single	o_3^6	$o_3, k=1$	1.0000
complete	pairwise	$o_3, k=5$	0.4306
complete	o_3^4	$o_3, k=5$	0.3050
complete	o_3^6	$o_3, k=5$	0.3050
average	pairwise	$o_3, k=5$	0.6033
average	o_3^4	$o_3, k=5$	0.6075
average	o_3^6	$o_3, k=5$	0.6075

single	pairwise	$o_3, k=5$	0.5067
single	o_3^4	$o_3, k=5$	0.6557
single	o_3^6	$o_3, k=5$	0.6557
complete	pairwise	$o_3, k=10$	0.5203
complete	o_3^4	$o_3, k=10$	0.3400
complete	o_3^6	$o_3, k=10$	0.3400
average	pairwise	$o_3, k=10$	0.6183
average	o_3^4	$o_3, k=10$	0.5625
average	o_3^6	$o_3, k=10$	0.5625
single	pairwise	$o_3, k=10$	0.2950
single	o_3^4	$o_3, k=10$	0.4229
single	o_3^6	$o_3, k=10$	0.4216
complete	pairwise	$o_4, k=1$	1.0000
complete	o_4^4	$o_4, k=1$	0.6667
complete	o_4^6	$o_4, k=1$	0.6667
average	pairwise	$o_4, k=1$	1.0000
average	o_4^4	$o_4, k=1$	0.6667
average	o_4^6	$o_4, k=1$	0.6667
single	pairwise	$o_4, k=1$	0.0833
single	o_4^4	$o_4, k=1$	0.2500
single	o_4^6	$o_4, k=1$	0.2500
complete	pairwise	$o_4, k=5$	0.2533
complete	o_4^4	$o_4, k=5$	0.1867
complete	o_4^6	$o_4, k=5$	0.1867
average	pairwise	$o_4, k=5$	0.3778
average	o_4^4	$o_4, k=5$	0.2667
average	o_4^6	$o_4, k=5$	0.3333
single	pairwise	$o_4, k=5$	0.1111
single	o_4^4	$o_4, k=5$	0.2500
single	o_4^6	$o_4, k=5$	0.2500
complete	pairwise	$o_4, k=10$	0.4169
complete	o_4^4	$o_4, k=10$	0.4533
complete	o_4^6	$o_4, k=10$	0.4533
average	pairwise	$o_4, k=10$	0.4803
average	o_4^4	$o_4, k=10$	0.4933

average	o_4^6	$o_4, k=10$	0.5267
single	pairwise	$o_4, k=10$	0.0972
single	o_4^4	$o_4, k=10$	0.2183
single	o_4^6	$o_4, k=10$	0.2412

Table A-2: Clustering metrics with bibliographic coupling for "Adaptive Optics" data set. Red and black text marks standard-versus-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	0.7500
complete	o_3^4	$o_3, k=1$	0.7500
average	pairwise	$o_3, k=1$	0.7500
average	o_3^4	$o_3, k=1$	0.7500
single	pairwise	$o_3, k=1$	0.7500
single	o_3^4	$o_3, k=1$	1.0000
complete	pairwise	$o_3, k=5$	0.4700
complete	o_3^4	$o_3, k=5$	0.5800
average	pairwise	$o_3, k=5$	0.6000
average	o_3^4	$o_3, k=5$	0.6557
single	pairwise	$o_3, k=5$	0.5167
single	o_3^4	$o_3, k=5$	0.6857
complete	pairwise	$o_3, k=10$	0.3750
complete	o_3^4	$o_3, k=10$	0.4400
average	pairwise	$o_3, k=10$	0.5028
average	o_3^4	$o_3, k=10$	0.5704
single	pairwise	$o_3, k=10$	0.4500
single	o_3^4	$o_3, k=10$	0.5926
complete	pairwise	$o_4, k=1$	1.0000
complete	o_4^4	$o_4, k=1$	0.0667
complete	o_4^6	$o_4, k=1$	1.0000
average	pairwise	$o_4, k=1$	1.0000
average	o_4^4	$o_4, k=1$	0.3333
average	o_4^6	$o_4, k=1$	1.0000
single	pairwise	$o_4, k=1$	1.0000
single	o_4^4	$o_4, k=1$	0.3333
single	o_4^6	$o_4, k=1$	1.0000
complete	pairwise	$o_4, k=5$	0.2533
complete	o_4^4	$o_4, k=5$	0.3333
complete	o_4^6	$o_4, k=5$	0.2533

average	pairwise	$o_4, k=5$	0.3882
average	o_4^4	$o_4, k=5$	0.3867
average	o_4^6	$o_4, k=5$	0.4667
single	pairwise	$o_4, k=5$	0.3778
single	o_4^4	$o_4, k=5$	0.3576
single	o_4^6	$o_4, k=5$	0.4667
complete	pairwise	$o_4, k=10$	0.1600
complete	o_4^4	$o_4, k=10$	0.2667
complete	o_4^6	$o_4, k=10$	0.1600
average	pairwise	$o_4, k=10$	0.3714
average	o_4^4	$o_4, k=10$	0.3907
average	o_4^6	$o_4, k=10$	0.3774
single	pairwise	$o_4, k=10$	0.3000
single	o_4^4	$o_4, k=10$	0.3566
single	o_4^6	$o_4, k=10$	0.4320

Table A-3: Clustering metrics for "Collagen" data set. Red and black text marks standard-versus-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	0.6000
complete	o_3^4	$o_3, k=1$	1.0000
average	pairwise	$o_3, k=1$	0.6000
average	o_3^4	$o_3, k=1$	1.0000
single	pairwise	$o_3, k=1$	0.3750
single	o_3^4	$o_3, k=1$	1.0000
complete	pairwise	$o_3, k=5$	0.5713
complete	o_3^4	$o_3, k=5$	0.7557
average	pairwise	$o_3, k=5$	0.7100
average	o_3^4	$o_3, k=5$	0.7700
single	pairwise	$o_3, k=5$	0.5450
single	o_3^4	$o_3, k=5$	0.7900
complete	pairwise	$o_3, k=10$	0.5990
complete	o_3^4	$o_3, k=10$	0.6854
average	pairwise	$o_3, k=10$	0.6779
average	o_3^4	$o_3, k=10$	0.7050
single	pairwise	$o_3, k=10$	0.4600
single	o_3^4	$o_3, k=10$	0.7150
complete	pairwise	$o_4, k=1$	0.8000
complete	o_4^4	$o_4, k=1$	1.0000
average	pairwise	$o_4, k=1$	0.8000
average	o_4^4	$o_4, k=1$	1.0000
single	pairwise	$o_4, k=1$	0.5000
single	o_4^4	$o_4, k=1$	1.0000
complete	pairwise	$o_4, k=5$	0.5867
complete	o_4^4	$o_4, k=5$	0.6429
average	pairwise	$o_4, k=5$	0.6629
average	o_4^4	$o_4, k=5$	0.7000
single	pairwise	$o_4, k=5$	0.3000
single	o_4^4	$o_4, k=5$	0.7000
complete	pairwise	$o_4, k=10$	0.5511

complete	o_4^4	$o_4, k=10$	0.5873
average	pairwise	$o_4, k=10$	0.6424
average	o_4^4	$o_4, k=10$	0.6533
single	pairwise	$o_4, k=10$	0.3000
single	o_4^4	$o_4, k=10$	0.6533

Table A-4: Clustering metrics for "Genetic Algorithms and Neural Networks" data set. Red and black text marks standard-versus-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	0.0526
complete	o_3^4	$o_3, k=1$	0.0526
complete	o_3^6	$o_3, k=1$	1.0000
average	pairwise	$o_3, k=1$	0.5000
average	o_3^4	$o_3, k=1$	0.2143
average	o_3^6	$o_3, k=1$	1.0000
single	pairwise	$o_3, k=1$	0.3000
single	o_3^4	$o_3, k=1$	0.7500
single	o_3^6	$o_3, k=1$	1.0000
complete	pairwise	$o_3, k=5$	0.1088
complete	o_3^4	$o_3, k=5$	0.4105
complete	o_3^6	$o_3, k=5$	0.6800
average	pairwise	$o_3, k=5$	0.4771
average	o_3^4	$o_3, k=5$	0.4429
average	o_3^6	$o_3, k=5$	0.6800
single	pairwise	$o_3, k=5$	0.1350
single	o_3^4	$o_3, k=5$	0.4500
single	o_3^6	$o_3, k=5$	0.6800
complete	pairwise	$o_3, k=10$	0.2035
complete	o_3^4	$o_3, k=10$	0.5053
complete	o_3^6	$o_3, k=10$	0.6800
average	pairwise	$o_3, k=10$	0.2035
average	o_3^4	$o_3, k=10$	0.5214
average	o_3^6	$o_3, k=10$	0.6800
single	pairwise	$o_3, k=10$	0.1144
single	o_3^4	$o_3, k=10$	0.3964
single	o_3^6	$o_3, k=10$	0.4214
complete	pairwise	$o_4, k=1$	0.0702
complete	o_4^4	$o_4, k=1$	0.6667

average	pairwise	$o_4, k=1$	0.5714
average	o_4^4	$o_4, k=1$	0.6667
single	pairwise	$o_4, k=1$	0.1250
single	o_4^4	$o_4, k=1$	0.6667
complete	pairwise	$o_4, k=5$	0.1450
complete	o_4^4	$o_4, k=5$	0.6667
average	pairwise	$o_4, k=5$	0.6171
average	o_4^4	$o_4, k=5$	0.6667
single	pairwise	$o_4, k=5$	0.1250
single	o_4^4	$o_4, k=5$	0.6667
complete	pairwise	$o_4, k=10$	0.1450
complete	o_4^4	$o_4, k=10$	0.6333
average	pairwise	$o_4, k=10$	0.4286
average	o_4^4	$o_4, k=10$	0.6444
single	pairwise	$o_4, k=10$	0.1250
single	o_4^4	$o_4, k=10$	0.6444

Table A-5: Clustering metrics for "Quantum Gravity and Strings" data set. Red and black text marks standard-versus-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	1.0000
complete	o_3^4	$o_3, k=1$	1.0000
average	pairwise	$o_3, k=1$	1.0000
average	o_3^4	$o_3, k=1$	1.0000
single	pairwise	$o_3, k=1$	1.0000
single	o_3^4	$o_3, k=1$	1.0000
complete	pairwise	$o_3, k=5$	0.9000
complete	o_3^4	$o_3, k=5$	0.9500
average	pairwise	$o_3, k=5$	0.9000
average	o_3^4	$o_3, k=5$	0.9500
single	pairwise	$o_3, k=5$	0.9000
single	o_3^4	$o_3, k=5$	0.9500
complete	pairwise	$o_3, k=10$	0.7123
complete	o_3^4	$o_3, k=10$	0.7545
average	pairwise	$o_3, k=10$	0.7425
average	o_3^4	$o_3, k=10$	0.7975
single	pairwise	$o_3, k=10$	0.6857
single	o_3^4	$o_3, k=10$	0.8029
complete	pairwise	$o_4, k=1$	0.6667
complete	o_4^4	$o_4, k=1$	1.0000
average	pairwise	$o_4, k=1$	0.8000
average	o_4^4	$o_4, k=1$	1.0000
single	pairwise	$o_4, k=1$	0.5714
single	o_4^4	$o_4, k=1$	1.0000
complete	pairwise	$o_4, k=5$	0.7394
complete	o_4^4	$o_4, k=5$	0.8061
average	pairwise	$o_4, k=5$	0.7933
average	o_4^4	$o_4, k=5$	0.7476
single	pairwise	$o_4, k=5$	0.7286
single	o_4^4	$o_4, k=5$	0.7822

complete	pairwise	$o_4, k = 10$	0.7030
complete	o_4^4	$o_4, k = 10$	0.7364
average	pairwise	$o_4, k = 10$	0.7433
average	o_4^4	$o_4, k = 10$	0.7205
single	pairwise	$o_4, k = 10$	0.6500
single	o_4^4	$o_4, k = 10$	0.7378

Table A-6: Clustering metrics with bibliographic coupling for "Quantum Gravity and Strings" data set. Red and black text marks standard-versus-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	0.3000
complete	o_3^4	$o_3, k=1$	0.3750
average	pairwise	$o_3, k=1$	1.0000
average	o_3^4	$o_3, k=1$	1.0000
single	pairwise	$o_3, k=1$	1.0000
single	o_3^4	$o_3, k=1$	1.0000
complete	pairwise	$o_3, k=5$	0.5800
complete	o_3^4	$o_3, k=5$	0.6250
average	pairwise	$o_3, k=5$	0.7357
average	o_3^4	$o_3, k=5$	0.7700
single	pairwise	$o_3, k=5$	0.7214
single	o_3^4	$o_3, k=5$	0.7700
complete	pairwise	$o_3, k=10$	0.4850
complete	o_3^4	$o_3, k=10$	0.5000
average	pairwise	$o_3, k=10$	0.6636
average	o_3^4	$o_3, k=10$	0.6779
single	pairwise	$o_3, k=10$	0.6393
single	o_3^4	$o_3, k=10$	0.6779
complete	pairwise	$o_4, k=1$	0.4000
complete	o_4^4	$o_4, k=1$	0.0800
complete	o_4^6	$o_4, k=1$	0.5000
average	pairwise	$o_4, k=1$	0.5714
average	o_4^4	$o_4, k=1$	0.6667
average	o_4^6	$o_4, k=1$	0.6667
single	pairwise	$o_4, k=1$	0.5714
single	o_4^4	$o_4, k=1$	0.6667
single	o_4^6	$o_4, k=1$	0.6667
complete	pairwise	$o_4, k=5$	0.3360
complete	o_4^4	$o_4, k=5$	0.2640

complete	o_4^6	$o_4, k=5$	0.5889
average	pairwise	$o_4, k=5$	0.5152
average	o_4^4	$o_4, k=5$	0.5933
average	o_4^6	$o_4, k=5$	0.6156
single	pairwise	$o_4, k=5$	0.4962
single	o_4^4	$o_4, k=5$	0.6137
single	o_4^6	$o_4, k=5$	0.6210
complete	pairwise	$o_4, k=10$	0.3680
complete	o_4^4	$o_4, k=10$	0.2893
complete	o_4^6	$o_4, k=10$	0.5744
average	pairwise	$o_4, k=10$	0.6052
average	o_4^4	$o_4, k=10$	0.6600
average	o_4^6	$o_4, k=10$	0.6711
single	pairwise	$o_4, k=10$	0.5767
single	o_4^4	$o_4, k=10$	0.6702
single	o_4^6	$o_4, k=10$	0.6738

Table A-7: Clustering metrics for "Wavelets (1-100)" data set. Red and black text marks standard-vs-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	0.0882
complete	o_3^4	$o_3, k=1$	1.0000
average	pairwise	$o_3, k=1$	1.0000
average	o_3^4	$o_3, k=1$	1.0000
single	pairwise	$o_3, k=1$	1.0000
single	o_3^4	$o_3, k=1$	1.0000
complete	pairwise	$o_3, k=5$	0.5891
complete	o_3^4	$o_3, k=5$	0.8500
average	pairwise	$o_3, k=5$	0.7200
average	o_3^4	$o_3, k=5$	0.8500
single	pairwise	$o_3, k=5$	0.5500
single	o_3^4	$o_3, k=5$	0.8500
complete	pairwise	$o_3, k=10$	0.5374
complete	o_3^4	$o_3, k=10$	0.7000
average	pairwise	$o_3, k=10$	0.5900
average	o_3^4	$o_3, k=10$	0.7000
single	pairwise	$o_3, k=10$	0.3480
single	o_3^4	$o_3, k=10$	0.7000
complete	pairwise	$o_4, k=1$	0.5714
complete	o_4^4	$o_4, k=1$	1.0000
average	pairwise	$o_4, k=1$	0.4000
average	o_4^4	$o_4, k=1$	1.0000
single	pairwise	$o_4, k=1$	0.3333
single	o_4^4	$o_4, k=1$	1.0000
complete	pairwise	$o_4, k=5$	0.5645
complete	o_4^4	$o_4, k=5$	0.8933
average	pairwise	$o_4, k=5$	0.6400
average	o_4^4	$o_4, k=5$	0.8933
single	pairwise	$o_4, k=5$	0.1947
single	o_4^4	$o_4, k=5$	0.8267
complete	pairwise	$o_4, k=10$	0.5940

complete	o_4^4	$o_4, k=10$	0.7584
average	pairwise	$o_4, k=10$	0.6354
average	o_4^4	$o_4, k=10$	0.7610
single	pairwise	$o_4, k=10$	0.1773
single	o_4^4	$o_4, k=10$	0.7100

Table A-8: Clustering metrics for "Wavelets (1-500)" data set. Red and black text marks standard-vs-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	1.0000
complete	o_3^4	$o_3, k=1$	1.0000
average	pairwise	$o_3, k=1$	1.0000
average	o_3^4	$o_3, k=1$	1.0000
single	pairwise	$o_3, k=1$	1.0000
single	o_3^4	$o_3, k=1$	1.0000
complete	pairwise	$o_3, k=5$	0.5722
complete	o_3^4	$o_3, k=5$	0.8700
average	pairwise	$o_3, k=5$	0.5972
average	o_3^4	$o_3, k=5$	0.8700
single	pairwise	$o_3, k=5$	0.4069
single	o_3^4	$o_3, k=5$	0.8700
complete	pairwise	$o_3, k=10$	0.5222
complete	o_3^4	$o_3, k=10$	0.7800
average	pairwise	$o_3, k=10$	0.5336
average	o_3^4	$o_3, k=10$	0.7800
single	pairwise	$o_3, k=10$	0.3609
single	o_3^4	$o_3, k=10$	0.7800
complete	pairwise	$o_4, k=1$	1.0000
complete	o_4^4	$o_4, k=1$	1.0000
average	pairwise	$o_4, k=1$	1.0000
average	o_4^4	$o_4, k=1$	1.0000
single	pairwise	$o_4, k=1$	0.2857
single	o_4^4	$o_4, k=1$	1.0000
complete	pairwise	$o_4, k=5$	0.2593
complete	o_4^4	$o_4, k=5$	0.8800
average	pairwise	$o_4, k=5$	0.3296
average	o_4^4	$o_4, k=5$	0.8800
single	pairwise	$o_4, k=5$	0.2684
single	o_4^4	$o_4, k=5$	0.8800
complete	pairwise	$o_4, k=10$	0.1667

complete	o_4^4	$o_4, k=10$	0.7867
average	pairwise	$o_4, k=10$	0.2407
average	o_4^4	$o_4, k=10$	0.7867
single	pairwise	$o_4, k=10$	0.2277
single	o_4^4	$o_4, k=10$	0.7867

Table A-9: Clustering metrics for "Wavelets and Brownian" data set. Red and black text marks standard-versus-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	1.0000
complete	o_3^4	$o_3, k=1$	1.0000
average	pairwise	$o_3, k=1$	1.0000
average	o_3^4	$o_3, k=1$	1.0000
single	pairwise	$o_3, k=1$	0.6000
single	o_3^4	$o_3, k=1$	1.0000
complete	pairwise	$o_3, k=5$	0.5600
complete	o_3^4	$o_3, k=5$	0.6857
average	pairwise	$o_3, k=5$	0.6857
average	o_3^4	$o_3, k=5$	0.7057
single	pairwise	$o_3, k=5$	0.5657
single	o_3^4	$o_3, k=5$	0.7057
complete	pairwise	$o_3, k=10$	0.5800
complete	o_3^4	$o_3, k=10$	0.6479
average	pairwise	$o_3, k=10$	0.6479
average	o_3^4	$o_3, k=10$	0.6379
single	pairwise	$o_3, k=10$	0.5314
single	o_3^4	$o_3, k=10$	0.6379
complete	pairwise	$o_4, k=1$	0.6667
complete	o_4^4	$o_4, k=1$	1.0000
average	pairwise	$o_4, k=1$	1.0000
average	o_4^4	$o_4, k=1$	1.0000
single	pairwise	$o_4, k=1$	0.8000
single	o_4^4	$o_4, k=1$	1.0000
complete	pairwise	$o_4, k=5$	0.6667
complete	o_4^4	$o_4, k=5$	0.8133
average	pairwise	$o_4, k=5$	0.7600
average	o_4^4	$o_4, k=5$	0.8133
single	pairwise	$o_4, k=5$	0.7543
single	o_4^4	$o_4, k=5$	0.8133

complete	pairwise	$o_4, k = 10$	0.6810
complete	o_4^4	$o_4, k = 10$	0.7343
average	pairwise	$o_4, k = 10$	0.6933
average	o_4^4	$o_4, k = 10$	0.7343
single	pairwise	$o_4, k = 10$	0.6514
single	o_4^4	$o_4, k = 10$	0.7343

Table A-10: Clustering metrics with bibliographic coupling for "Wavelets and Brownian" data set. Red and black text marks comparisons for standard versus hybrid distances.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	0.0508
complete	o_3^4	$o_3, k=1$	0.1071
average	pairwise	$o_3, k=1$	0.2500
average	o_3^4	$o_3, k=1$	0.5000
single	pairwise	$o_3, k=1$	0.2308
single	o_3^4	$o_3, k=1$	0.6000
complete	pairwise	$o_3, k=5$	0.3305
complete	o_3^4	$o_3, k=5$	0.5129
average	pairwise	$o_3, k=5$	0.5500
average	o_3^4	$o_3, k=5$	0.7200
single	pairwise	$o_3, k=5$	0.5923
single	o_3^4	$o_3, k=5$	0.7400
complete	pairwise	$o_3, k=10$	0.1907
complete	o_3^4	$o_3, k=10$	0.5714
average	pairwise	$o_3, k=10$	0.3125
average	o_3^4	$o_3, k=10$	0.6825
single	pairwise	$o_3, k=10$	0.4115
single	o_3^4	$o_3, k=10$	0.6825
complete	pairwise	$o_4, k=1$	0.0678
complete	o_4^4	$o_4, k=1$	1.0000
average	pairwise	$o_4, k=1$	0.1000
average	o_4^4	$o_4, k=1$	1.0000
single	pairwise	$o_4, k=1$	0.3077
single	o_4^4	$o_4, k=1$	1.0000
complete	pairwise	$o_4, k=5$	0.0678
complete	o_4^4	$o_4, k=5$	0.6450
average	pairwise	$o_4, k=5$	0.2400
average	o_4^4	$o_4, k=5$	0.7333
single	pairwise	$o_4, k=5$	0.3077

single	o_4^4	$o_4, k=5$	0.7533
complete	pairwise	$o_4, k=10$	0.0678
complete	o_4^4	$o_4, k=10$	0.5275
average	pairwise	$o_4, k=10$	0.1700
average	o_4^4	$o_4, k=10$	0.6600
single	pairwise	$o_4, k=10$	0.3077
single	o_4^4	$o_4, k=10$	0.6900

Appendix B

Clustering Metrics for Excluding Infrequent Itemsets

This appendix gives detailed results for the experiments conducted in Chapter 4, Section 4.1. In particular, it gives clustering metrics for each test case, along with the corresponding choice of experimental inputs. The results are presented Tables B-1 through B-5, corresponding to the 5 data sets employed.

Table B-1: Clustering metrics for hybrid distances with reduced computational complexity via *minsup*, for "Collagen" data set. Red and black text marks standard-versus-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	0.6000
complete	o_3^4	$o_3, k=1$	1.0000
complete	o_3^4, *minsup* = 2	$o_3, k=1$	1.0000
complete	o_3^4, *minsup* = 4	$o_3, k=1$	1.0000
average	pairwise	$o_3, k=1$	0.6000
average	o_3^4	$o_3, k=1$	1.0000
average	o_3^4, *minsup* = 2	$o_3, k=1$	1.0000
average	o_3^4, *minsup* = 4	$o_3, k=1$	1.0000
single	pairwise	$o_3, k=1$	0.3750
single	o_3^4	$o_3, k=1$	1.0000
single	o_3^4, *minsup* = 2	$o_3, k=1$	1.0000
single	o_3^4, *minsup* = 4	$o_3, k=1$	1.0000
complete	pairwise	$o_3, k=5$	0.5713

complete	o_3^4	$o_3, k = 5$	0.7557
complete	o_3^4, *minsup* = 2	$o_3, k = 5$	0.7557
complete	o_3^4, *minsup* = 4	$o_3, k = 5$	0.6813
average	pairwise	$o_3, k = 5$	0.7100
average	o_3^4	$o_3, k = 5$	0.7700
average	o_3^4, *minsup* = 2	$o_3, k = 5$	0.7700
average	o_3^4, *minsup* = 4	$o_3, k = 5$	0.7557
single	pairwise	$o_3, k = 5$	0.5450
single	o_3^4	$o_3, k = 5$	0.7900
single	o_3^4, *minsup* = 2	$o_3, k = 5$	0.7900
single	o_3^4, *minsup* = 4	$o_3, k = 5$	0.7900
complete	pairwise	$o_3, k = 10$	0.5990
complete	o_3^4	$o_3, k = 10$	0.6854
complete	o_3^4, *minsup* = 2	$o_3, k = 10$	0.6854
complete	o_3^4, *minsup* = 4	$o_3, k = 10$	0.6482
average	pairwise	$o_3, k = 10$	0.6779
average	o_3^4	$o_3, k = 10$	0.7050
average	o_3^4, *minsup* = 2	$o_3, k = 10$	0.7050
average	o_3^4, *minsup* = 4	$o_3, k = 10$	0.6979
single	pairwise	$o_3, k = 10$	0.4600
single	o_3^4	$o_3, k = 10$	0.7150
single	o_3^4, *minsup* = 2	$o_3, k = 10$	0.7150
single	o_3^4, *minsup* = 4	$o_3, k = 10$	0.7150
complete	pairwise	$o_4, k = 1$	0.8000
complete	o_4^4	$o_4, k = 1$	1.0000
complete	o_4^4, *minsup* = 2	$o_4, k = 1$	1.0000
complete	o_4^4, *minsup* = 4	$o_4, k = 1$	1.0000
average	pairwise	$o_4, k = 1$	0.8000
average	o_4^4	$o_4, k = 1$	1.0000
average	o_4^4, *minsup* = 2	$o_4, k = 1$	1.0000
average	o_4^4, *minsup* = 4	$o_4, k = 1$	1.0000
single	pairwise	$o_4, k = 1$	0.5000
single	o_4^4	$o_4, k = 1$	1.0000
single	o_4^4, *minsup* = 2	$o_4, k = 1$	1.0000

single	o_4^4, *minsup* = 4	$o_4, k = 1$	1.0000
complete	pairwise	$o_4, k = 5$	0.5867
complete	o_4^4	$o_4, k = 5$	0.6429
complete	o_4^4, *minsup* = 2	$o_4, k = 5$	0.6429
complete	o_4^4, *minsup* = 4	$o_4, k = 5$	0.6429
average	pairwise	$o_4, k = 5$	0.6629
average	o_4^4	$o_4, k = 5$	0.7000
average	o_4^4, *minsup* = 2	$o_4, k = 5$	0.7000
average	o_4^4, *minsup* = 4	$o_4, k = 5$	0.7000
single	pairwise	$o_4, k = 5$	0.3000
single	o_4^4	$o_4, k = 5$	0.7000
single	o_4^4, *minsup* = 2	$o_4, k = 5$	0.7000
single	o_4^4, *minsup* = 4	$o_4, k = 5$	0.7000
complete	pairwise	$o_4, k = 10$	0.5511
complete	o_4^4	$o_4, k = 10$	0.5873
complete	o_4^4, *minsup* = 2	$o_4, k = 10$	0.5873
complete	o_4^4, *minsup* = 4	$o_4, k = 10$	0.5873
average	pairwise	$o_4, k = 10$	0.6424
average	o_4^4	$o_4, k = 10$	0.6533
average	o_4^4, *minsup* = 2	$o_4, k = 10$	0.6533
average	o_4^4, *minsup* = 4	$o_4, k = 10$	0.6578
single	pairwise	$o_4, k = 10$	0.3000
single	o_4^4	$o_4, k = 10$	0.6533
single	o_4^4, *minsup* = 2	$o_4, k = 10$	0.6533
single	o_4^4, *minsup* = 4	$o_4, k = 10$	0.6578

Table B-2: Clustering metrics for hybrid distances with reduced computational complexity via *minsup*, for "Quantum Gravity and Strings" data set. Red and black text marks standard-vs-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	1.0000
complete	o_3^4	$o_3, k=1$	1.0000
complete	o_3^4, *minsup* = 2	$o_3, k=1$	1.0000
complete	o_3^4, *minsup* = 4	$o_3, k=1$	1.0000
average	pairwise	$o_3, k=1$	1.0000
average	o_3^4	$o_3, k=1$	1.0000
average	o_3^4, *minsup* = 2	$o_3, k=1$	1.0000
average	o_3^4, *minsup* = 4	$o_3, k=1$	1.0000
single	pairwise	$o_3, k=1$	1.0000
single	o_3^4	$o_3, k=1$	1.0000
single	o_3^4, *minsup* = 2	$o_3, k=1$	1.0000
single	o_3^4, *minsup* = 4	$o_3, k=1$	1.0000
complete	pairwise	$o_3, k=5$	0.9000
complete	o_3^4	$o_3, k=5$	0.9500
complete	o_3^4, *minsup* = 2	$o_3, k=5$	0.9500
complete	o_3^4, *minsup* = 4	$o_3, k=5$	0.9500
average	pairwise	$o_3, k=5$	0.9000
average	o_3^4	$o_3, k=5$	0.9500
average	o_3^4, *minsup* = 2	$o_3, k=5$	0.9500
average	o_3^4, *minsup* = 4	$o_3, k=5$	0.9500
single	pairwise	$o_3, k=5$	0.9000
single	o_3^4	$o_3, k=5$	0.9500
single	o_3^4, *minsup* = 2	$o_3, k=5$	0.9500
single	o_3^4, *minsup* = 4	$o_3, k=5$	0.9500
complete	pairwise	$o_3, k=10$	0.7123
complete	o_3^4	$o_3, k=10$	0.7545
complete	o_3^4, *minsup* = 2	$o_3, k=10$	0.7545
complete	o_3^4, *minsup* = 4	$o_3, k=10$	0.7120

average	pairwise	$o_3, k = 10$	0.7425
average	o_3^4	$o_3, k = 10$	0.7975
average	o_3^4, *minsup* = 2	$o_3, k = 10$	0.7975
average	o_3^4, *minsup* = 4	$o_3, k = 10$	0.7975
single	pairwise	$o_3, k = 10$	0.6857
single	o_3^4	$o_3, k = 10$	0.8029
single	o_3^4, *minsup* = 2	$o_3, k = 10$	0.8029
single	o_3^4, *minsup* = 4	$o_3, k = 10$	0.8029
complete	pairwise	$o_4, k = 1$	0.6667
complete	o_4^4	$o_4, k = 1$	1.0000
complete	o_4^4, *minsup* = 2	$o_4, k = 1$	1.0000
complete	o_4^4, *minsup* = 4	$o_4, k = 1$	1.0000
average	pairwise	$o_4, k = 1$	0.8000
average	o_4^4	$o_4, k = 1$	1.0000
average	o_4^4, *minsup* = 2	$o_4, k = 1$	1.0000
average	o_4^4, *minsup* = 4	$o_4, k = 1$	1.0000
single	pairwise	$o_4, k = 1$	0.5714
single	o_4^4	$o_4, k = 1$	1.0000
single	o_4^4, *minsup* = 2	$o_4, k = 1$	1.0000
single	o_4^4, *minsup* = 4	$o_4, k = 1$	1.0000
complete	pairwise	$o_4, k = 5$	0.7394
complete	o_4^4	$o_4, k = 5$	0.8061
complete	o_4^4, *minsup* = 2	$o_4, k = 5$	0.8061
complete	o_4^4, *minsup* = 4	$o_4, k = 5$	0.7493
average	pairwise	$o_4, k = 5$	0.7933
average	o_4^4	$o_4, k = 5$	0.7476
average	o_4^4, *minsup* = 2	$o_4, k = 5$	0.7476
average	o_4^4, *minsup* = 4	$o_4, k = 5$	0.7667
single	pairwise	$o_4, k = 5$	0.7286
single	o_4^4	$o_4, k = 5$	0.7822
single	o_4^4, *minsup* = 2	$o_4, k = 5$	0.7822
single	o_4^4, *minsup* = 4	$o_4, k = 5$	0.7667
complete	pairwise	$o_4, k = 10$	0.7030
complete	o_4^4	$o_4, k = 10$	0.7364
complete	o_4^4, *minsup* = 2	$o_4, k = 10$	0.7364

complete	o_4^4, *minsup* = 4	$o_4, k = 10$	0.7080
average	pairwise	$o_4, k = 10$	0.7433
average	o_4^4	$o_4, k = 10$	0.7205
average	o_4^4, *minsup* = 2	$o_4, k = 10$	0.7205
average	o_4^4, *minsup* = 4	$o_4, k = 10$	0.7300
single	pairwise	$o_4, k = 10$	0.6500
single	o_4^4	$o_4, k = 10$	0.7378
single	o_4^4, *minsup* = 2	$o_4, k = 10$	0.7378
single	o_4^4, *minsup* = 4	$o_4, k = 10$	0.7300

Table B-3: Clustering metrics for hybrid distances with reduced computational complexity via *minsup*, for "Wavelets (1-500)" data set. Red and black text marks standard-versus-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	1.0000
complete	o_3^4	$o_3, k=1$	1.0000
complete	o_3^4, *minsup* = 2	$o_3, k=1$	1.0000
complete	o_3^4, *minsup* = 4	$o_3, k=1$	1.0000
average	pairwise	$o_3, k=1$	1.0000
average	o_3^4	$o_3, k=1$	1.0000
average	o_3^4, *minsup* = 2	$o_3, k=1$	1.0000
average	o_3^4, *minsup* = 4	$o_3, k=1$	1.0000
single	pairwise	$o_3, k=1$	1.0000
single	o_3^4	$o_3, k=1$	1.0000
single	o_3^4, *minsup* = 2	$o_3, k=1$	1.0000
single	o_3^4, *minsup* = 4	$o_3, k=1$	1.0000
complete	pairwise	$o_3, k=5$	0.5722
complete	o_3^4	$o_3, k=5$	0.8700
complete	o_3^4, *minsup* = 2	$o_3, k=5$	0.8700
complete	o_3^4, *minsup* = 4	$o_3, k=5$	0.8700
average	pairwise	$o_3, k=5$	0.5972
average	o_3^4	$o_3, k=5$	0.8700
average	o_3^4, *minsup* = 2	$o_3, k=5$	0.8700
average	o_3^4, *minsup* = 4	$o_3, k=5$	0.8700
single	pairwise	$o_3, k=5$	0.4069
single	o_3^4	$o_3, k=5$	0.8700
single	o_3^4, *minsup* = 2	$o_3, k=5$	0.8700
single	o_3^4, *minsup* = 4	$o_3, k=5$	0.8700
complete	pairwise	$o_3, k=10$	0.5222
complete	o_3^4	$o_3, k=10$	0.7800
complete	o_3^4, *minsup* = 2	$o_3, k=10$	0.7800
complete	o_3^4, *minsup* = 4	$o_3, k=10$	0.7106

average	pairwise	$o_3, k=10$	0.5336
average	o_3^4	$o_3, k=10$	0.7800
average	o_3^4, *minsup* = 2	$o_3, k=10$	0.7800
average	o_3^4, *minsup* = 4	$o_3, k=10$	0.7800
single	pairwise	$o_3, k=10$	0.3609
single	o_3^4	$o_3, k=10$	0.7800
single	o_3^4, *minsup* = 2	$o_3, k=10$	0.7800
single	o_3^4, *minsup* = 4	$o_3, k=10$	0.7800
complete	pairwise	$o_4, k=1$	1.0000
complete	o_4^4	$o_4, k=1$	1.0000
complete	o_4^4, *minsup* = 2	$o_4, k=1$	1.0000
complete	o_4^4, *minsup* = 4	$o_4, k=1$	0.8000
average	pairwise	$o_4, k=1$	1.0000
average	o_4^4	$o_4, k=1$	1.0000
average	o_4^4, *minsup* = 2	$o_4, k=1$	1.0000
average	o_4^4, *minsup* = 4	$o_4, k=1$	0.8000
single	pairwise	$o_4, k=1$	0.2857
single	o_4^4	$o_4, k=1$	1.0000
single	o_4^4, *minsup* = 2	$o_4, k=1$	1.0000
single	o_4^4, *minsup* = 4	$o_4, k=1$	0.8000
complete	pairwise	$o_4, k=5$	0.2593
complete	o_4^4	$o_4, k=5$	0.8800
complete	o_4^4, *minsup* = 2	$o_4, k=5$	0.8800
complete	o_4^4, *minsup* = 4	$o_4, k=5$	0.8800
average	pairwise	$o_4, k=5$	0.3296
average	o_4^4	$o_4, k=5$	0.8800
average	o_4^4, *minsup* = 2	$o_4, k=5$	0.8800
average	o_4^4, *minsup* = 4	$o_4, k=5$	0.8800
single	pairwise	$o_4, k=5$	0.2684
single	o_4^4	$o_4, k=5$	0.8800
single	o_4^4, *minsup* = 2	$o_4, k=5$	0.8800
single	o_4^4, *minsup* = 4	$o_4, k=5$	0.8400
complete	pairwise	$o_4, k=10$	0.1667
complete	o_4^4	$o_4, k=10$	0.7867
complete	o_4^4, *minsup* = 2	$o_4, k=10$	0.7867

complete	o_4^4, *minsup* = 4	$o_4, k = 10$	0.5496
average	pairwise	$o_4, k = 10$	0.2407
average	o_4^4	$o_4, k = 10$	0.7867
average	o_4^4, *minsup* = 2	$o_4, k = 10$	0.7867
average	o_4^4, *minsup* = 4	$o_4, k = 10$	0.5496
single	pairwise	$o_4, k = 10$	0.2277
single	o_4^4	$o_4, k = 10$	0.7867
single	o_4^4, *minsup* = 2	$o_4, k = 10$	0.7867
single	o_4^4, *minsup* = 4	$o_4, k = 10$	0.5296

Table B-4: Clustering metrics for hybrid distances with reduced computational complexity via *minsup*, for "Wavelets and Brownian" data set. Red and black text marks standard-vs-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	1.0000
complete	o_3^4	$o_3, k=1$	1.0000
complete	o_3^4, *minsup* = 2	$o_3, k=1$	1.0000
complete	o_3^4, *minsup* = 4	$o_3, k=1$	1.0000
complete	o_3^4, *minsup* = 8	$o_3, k=1$	1.0000
average	pairwise	$o_3, k=1$	1.0000
average	o_3^4	$o_3, k=1$	1.0000
average	o_3^4, *minsup* = 2	$o_3, k=1$	1.0000
average	o_3^4, *minsup* = 4	$o_3, k=1$	1.0000
average	o_3^4, *minsup* = 8	$o_3, k=1$	1.0000
single	pairwise	$o_3, k=1$	0.6000
single	o_3^4	$o_3, k=1$	1.0000
single	o_3^4, *minsup* = 2	$o_3, k=1$	1.0000
single	o_3^4, *minsup* = 4	$o_3, k=1$	1.0000
single	o_3^4, *minsup* = 8	$o_3, k=1$	1.0000
complete	pairwise	$o_3, k=5$	0.5600
complete	o_3^4	$o_3, k=5$	0.6857
complete	o_3^4, *minsup* = 2	$o_3, k=5$	0.6857
complete	o_3^4, *minsup* = 4	$o_3, k=5$	0.6857
complete	o_3^4, *minsup* = 8	$o_3, k=5$	0.7057
average	pairwise	$o_3, k=5$	0.6857
average	o_3^4	$o_3, k=5$	0.7057
average	o_3^4, *minsup* = 2	$o_3, k=5$	0.7057
average	o_3^4, *minsup* = 4	$o_3, k=5$	0.7057
average	o_3^4, *minsup* = 8	$o_3, k=5$	0.7057
single	pairwise	$o_3, k=5$	0.5657
single	o_3^4	$o_3, k=5$	0.7057
single	o_3^4, *minsup* = 2	$o_3, k=5$	0.7057

single	o_3^4, *minsup* = 4	$o_3, k = 5$	0.7057
single	o_3^4, *minsup* = 8	$o_3, k = 5$	0.7057
complete	pairwise	$o_3, k = 10$	0.5800
complete	o_3^4	$o_3, k = 10$	0.6479
complete	o_3^4, *minsup* = 2	$o_3, k = 10$	0.6479
complete	o_3^4, *minsup* = 4	$o_3, k = 10$	0.6479
complete	o_3^4, *minsup* = 8	$o_3, k = 10$	0.6379
average	pairwise	$o_3, k = 10$	0.6479
average	o_3^4	$o_3, k = 10$	0.6379
average	o_3^4, *minsup* = 2	$o_3, k = 10$	0.6379
average	o_3^4, *minsup* = 4	$o_3, k = 10$	0.6379
average	o_3^4, *minsup* = 8	$o_3, k = 10$	0.6379
single	pairwise	$o_3, k = 10$	0.5314
single	o_3^4	$o_3, k = 10$	0.6379
single	o_3^4, *minsup* = 2	$o_3, k = 10$	0.6379
single	o_3^4, *minsup* = 4	$o_3, k = 10$	0.6379
single	o_3^4, *minsup* = 8	$o_3, k = 10$	0.6379
complete	pairwise	$o_4, k = 1$	0.6667
complete	o_4^4	$o_4, k = 1$	1.0000
complete	o_4^4, *minsup* = 2	$o_4, k = 1$	1.0000
complete	o_4^4, *minsup* = 4	$o_4, k = 1$	1.0000
average	pairwise	$o_4, k = 1$	1.0000
average	o_4^4	$o_4, k = 1$	1.0000
average	o_4^4, *minsup* = 2	$o_4, k = 1$	1.0000
average	o_4^4, *minsup* = 4	$o_4, k = 1$	1.0000
single	pairwise	$o_4, k = 1$	0.8000
single	o_4^4	$o_4, k = 1$	1.0000
single	o_4^4, *minsup* = 2	$o_4, k = 1$	1.0000
single	o_4^4, *minsup* = 4	$o_4, k = 1$	1.0000
complete	pairwise	$o_4, k = 5$	0.6667
complete	o_4^4	$o_4, k = 5$	0.8133
complete	o_4^4, *minsup* = 2	$o_4, k = 5$	0.8133
complete	o_4^4, *minsup* = 4	$o_4, k = 5$	0.8133
average	pairwise	$o_4, k = 5$	0.7600

average	o_4^4	$o_4, k = 5$	0.8133
average	o_4^4, *minsup* = 2	$o_4, k = 5$	0.8133
average	o_4^4, *minsup* = 4	$o_4, k = 5$	0.8133
single	pairwise	$o_4, k = 5$	0.7543
single	o_4^4	$o_4, k = 5$	0.8133
single	o_4^4, *minsup* = 2	$o_4, k = 5$	0.8133
single	o_4^4, *minsup* = 4	$o_4, k = 5$	0.8133
complete	pairwise	$o_4, k = 10$	0.6810
complete	o_4^4	$o_4, k = 10$	0.7343
complete	o_4^4, *minsup* = 2	$o_4, k = 10$	0.7343
complete	o_4^4, *minsup* = 4	$o_4, k = 10$	0.7343
average	pairwise	$o_4, k = 10$	0.6933
average	o_4^4	$o_4, k = 10$	0.7343
average	o_4^4, *minsup* = 2	$o_4, k = 10$	0.7343
average	o_4^4, *minsup* = 4	$o_4, k = 10$	0.7343
single	pairwise	$o_4, k = 10$	0.6514
single	o_4^4	$o_4, k = 10$	0.7343
single	o_4^4, *minsup* = 2	$o_4, k = 10$	0.7343
single	o_4^4, *minsup* = 4	$o_4, k = 10$	0.7343

Table B-5: Clustering metrics for hybrid distances with reduced computational complexity via *minsup*, for "Wavelets and Brownian" data set with bibliographic coupling. Red and black text marks standard-versus-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	0.0508
complete	o_3^4	$o_3, k=1$	0.1071
complete	o_3^4, *minsup* = 2	$o_3, k=1$	0.0508
complete	o_3^4, *minsup* = 4	$o_3, k=1$	0.0508
average	pairwise	$o_3, k=1$	0.2500
average	o_3^4	$o_3, k=1$	0.5000
average	o_3^4, *minsup* = 2	$o_3, k=1$	0.5000
average	o_3^4, *minsup* = 4	$o_3, k=1$	0.5000
single	pairwise	$o_3, k=1$	0.2308
single	o_3^4	$o_3, k=1$	0.6000
single	o_3^4, *minsup* = 2	$o_3, k=1$	0.6000
single	o_3^4, *minsup* = 4	$o_3, k=1$	0.5000
complete	pairwise	$o_3, k=5$	0.3305
complete	o_3^4	$o_3, k=5$	0.5129
complete	o_3^4, *minsup* = 2	$o_3, k=5$	0.4903
complete	o_3^4, *minsup* = 4	$o_3, k=5$	0.4903
average	pairwise	$o_3, k=5$	0.5500
average	o_3^4	$o_3, k=5$	0.7200
average	o_3^4, *minsup* = 2	$o_3, k=5$	0.7200
average	o_3^4, *minsup* = 4	$o_3, k=5$	0.7200
single	pairwise	$o_3, k=5$	0.5923
single	o_3^4	$o_3, k=5$	0.7400
single	o_3^4, *minsup* = 2	$o_3, k=5$	0.7400
single	o_3^4, *minsup* = 4	$o_3, k=5$	0.7200
complete	pairwise	$o_3, k=10$	0.1907
complete	o_3^4	$o_3, k=10$	0.5714
complete	o_3^4, *minsup* = 2	$o_3, k=10$	0.5602

complete	o_3^4, *minsup* = 4	$o_3, k = 10$	0.5353
average	pairwise	$o_3, k = 10$	0.3125
average	o_3^4	$o_3, k = 10$	0.6825
average	o_3^4, *minsup* = 2	$o_3, k = 10$	0.6825
average	o_3^4, *minsup* = 4	$o_3, k = 10$	0.6783
single	pairwise	$o_3, k = 10$	0.4115
single	o_3^4	$o_3, k = 10$	0.6825
single	o_3^4, *minsup* = 2	$o_3, k = 10$	0.6825
single	o_3^4, *minsup* = 4	$o_3, k = 10$	0.6825
complete	pairwise	$o_4, k = 1$	0.0678
complete	o_4^4	$o_4, k = 1$	1.0000
complete	o_4^4, *minsup* = 2	$o_4, k = 1$	1.0000
complete	o_4^4, *minsup* = 4	$o_4, k = 1$	1.0000
average	pairwise	$o_4, k = 1$	0.1000
average	o_4^4	$o_4, k = 1$	1.0000
average	o_4^4, *minsup* = 2	$o_4, k = 1$	1.0000
average	o_4^4, *minsup* = 4	$o_4, k = 1$	1.0000
single	pairwise	$o_4, k = 1$	0.3077
single	o_4^4	$o_4, k = 1$	1.0000
single	o_4^4, *minsup* = 2	$o_4, k = 1$	1.0000
single	o_4^4, *minsup* = 4	$o_4, k = 1$	1.0000
complete	pairwise	$o_4, k = 5$	0.0678
complete	o_4^4	$o_4, k = 5$	0.6450
complete	o_4^4, *minsup* = 2	$o_4, k = 5$	0.6336
complete	o_4^4, *minsup* = 4	$o_4, k = 5$	0.6336
average	pairwise	$o_4, k = 5$	0.2400
average	o_4^4	$o_4, k = 5$	0.7333
average	o_4^4, *minsup* = 2	$o_4, k = 5$	0.7261
average	o_4^4, *minsup* = 4	$o_4, k = 5$	0.7422
single	pairwise	$o_4, k = 5$	0.3077
single	o_4^4	$o_4, k = 5$	0.7533
single	o_4^4, *minsup* = 2	$o_4, k = 5$	0.7533
single	o_4^4, *minsup* = 4	$o_4, k = 5$	0.7533
complete	pairwise	$o_4, k = 10$	0.0678
complete	o_4^4	$o_4, k = 10$	0.5275

complete	o_4^4, *minsup* = 2	$o_4, k = 10$	0.5103
complete	o_4^4, *minsup* = 4	$o_4, k = 10$	0.5103
average	pairwise	$o_4, k = 10$	0.1700
average	o_4^4	$o_4, k = 10$	0.6600
average	o_4^4, *minsup* = 2	$o_4, k = 10$	0.6491
average	o_4^4, *minsup* = 4	$o_4, k = 10$	0.6733
single	pairwise	$o_4, k = 10$	0.3077
single	o_4^4	$o_4, k = 10$	0.6900
single	o_4^4, *minsup* = 2	$o_4, k = 10$	0.6900
single	o_4^4, *minsup* = 4	$o_4, k = 10$	0.6900

Appendix C

Clustering Metrics for Transaction and Item Weighting

This appendix gives detailed results for the experiments conducted in Chapter 4, Section 4.3. In particular, it gives clustering metrics for each test case, along with the corresponding choice of experimental inputs. The results are presented Tables C-1 through C-5, corresponding to the 5 data sets employed.

Table C-1: Clustering metrics for transaction and item weighting, "Collagen" data set. Red and black text marks standard-versus-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	0.6000
complete	transaction weight	$o_3, k=1$	0.7500
complete	item weight	$o_3, k=1$	0.4286
average	pairwise	$o_3, k=1$	0.6000
average	transaction weight	$o_3, k=1$	0.7500
average	item weight	$o_3, k=1$	0.1250
single	pairwise	$o_3, k=1$	0.3750
single	transaction weight	$o_3, k=1$	0.7500
single	item weight	$o_3, k=1$	0.1875
complete	pairwise	$o_3, k=5$	0.5713
complete	transaction weight	$o_3, k=5$	0.6394
complete	item weight	$o_3, k=5$	0.6714
average	pairwise	$o_3, k=5$	0.7100

average	transaction weight	$o_3, k = 5$	0.7400
average	item weight	$o_3, k = 5$	0.5850
single	pairwise	$o_3, k = 5$	0.5450
single	transaction weight	$o_3, k = 5$	0.6210
single	item weight	$o_3, k = 5$	0.1755
complete	pairwise	$o_3, k = 10$	0.5990
complete	transaction weight	$o_3, k = 10$	0.6197
complete	item weight	$o_3, k = 10$	0.5321
average	pairwise	$o_3, k = 10$	0.6779
average	transaction weight	$o_3, k = 10$	0.6829
average	item weight	$o_3, k = 10$	0.3550
single	pairwise	$o_3, k = 10$	0.4600
single	transaction weight	$o_3, k = 10$	0.5962
single	item weight	$o_3, k = 10$	0.1764
complete	pairwise	$o_4, k = 1$	0.8000
complete	transaction weight	$o_4, k = 1$	1.0000
complete	item weight	$o_4, k = 1$	0.5714
average	pairwise	$o_4, k = 1$	0.8000
average	transaction weight	$o_4, k = 1$	1.0000
average	item weight	$o_4, k = 1$	0.1667
single	pairwise	$o_4, k = 1$	0.5000
single	transaction weight	$o_4, k = 1$	1.0000
single	item weight	$o_4, k = 1$	0.2500
complete	pairwise	$o_4, k = 5$	0.5867
complete	transaction weight	$o_4, k = 5$	0.6000
complete	item weight	$o_4, k = 5$	0.4286
average	pairwise	$o_4, k = 5$	0.6629
average	transaction weight	$o_4, k = 5$	0.7029
average	item weight	$o_4, k = 5$	0.4095
single	pairwise	$o_4, k = 5$	0.3000
single	transaction weight	$o_4, k = 5$	0.5810
single	item weight	$o_4, k = 5$	0.2043
complete	pairwise	$o_4, k = 10$	0.5511
complete	transaction weight	$o_4, k = 10$	0.5400
complete	item weight	$o_4, k = 10$	0.4476
average	pairwise	$o_4, k = 10$	0.6424

average	transaction weight	$o_4, k = 10$	0.6524
average	item weight	$o_4, k = 10$	0.4619
single	pairwise	$o_4, k = 10$	0.3000
single	transaction weight	$o_4, k = 10$	0.5317
single	item weight	$o_4, k = 10$	0.1983

Table C-2: Clustering metrics for transaction and item weighting, "Quantum Gravity and Strings" data set. Red and black text marks standard-vs-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	1.0000
complete	transaction weight	$o_3, k=1$	1.0000
complete	item weight	$o_3, k=1$	1.0000
average	pairwise	$o_3, k=1$	1.0000
average	transaction weight	$o_3, k=1$	1.0000
average	item weight	$o_3, k=1$	1.0000
single	pairwise	$o_3, k=1$	1.0000
single	transaction weight	$o_3, k=1$	1.0000
single	item weight	$o_3, k=1$	1.0000
complete	pairwise	$o_3, k=5$	0.9000
complete	transaction weight	$o_3, k=5$	0.8545
complete	item weight	$o_3, k=5$	0.9500
average	pairwise	$o_3, k=5$	0.9000
average	transaction weight	$o_3, k=5$	0.9500
average	item weight	$o_3, k=5$	0.9500
single	pairwise	$o_3, k=5$	0.9000
single	transaction weight	$o_3, k=5$	0.9500
single	item weight	$o_3, k=5$	0.8700
complete	pairwise	$o_3, k=10$	0.7123
complete	transaction weight	$o_3, k=10$	0.6474
complete	item weight	$o_3, k=10$	0.7825
average	pairwise	$o_3, k=10$	0.7425
average	transaction weight	$o_3, k=10$	0.7475
average	item weight	$o_3, k=10$	0.7575
single	pairwise	$o_3, k=10$	0.6857
single	transaction weight	$o_3, k=10$	0.7475
single	item weight	$o_3, k=10$	0.6650
complete	pairwise	$o_4, k=1$	0.6667
complete	transaction weight	$o_4, k=1$	0.5000
complete	item weight	$o_4, k=1$	0.8000

average	pairwise	$o_4, k=1$	0.8000
average	transaction weight	$o_4, k=1$	0.6667
average	item weight	$o_4, k=1$	0.6667
single	pairwise	$o_4, k=1$	0.5714
single	transaction weight	$o_4, k=1$	0.6667
single	item weight	$o_4, k=1$	0.1667
complete	pairwise	$o_4, k=5$	0.7394
complete	transaction weight	$o_4, k=5$	0.5870
complete	item weight	$o_4, k=5$	0.7933
average	pairwise	$o_4, k=5$	0.7933
average	transaction weight	$o_4, k=5$	0.7667
average	item weight	$o_4, k=5$	0.7476
single	pairwise	$o_4, k=5$	0.7286
single	transaction weight	$o_4, k=5$	0.8267
single	item weight	$o_4, k=5$	0.5200
complete	pairwise	$o_4, k=10$	0.7030
complete	transaction weight	$o_4, k=10$	0.5435
complete	item weight	$o_4, k=10$	0.7433
average	pairwise	$o_4, k=10$	0.7433
average	transaction weight	$o_4, k=10$	0.7167
average	item weight	$o_4, k=10$	0.6690
single	pairwise	$o_4, k=10$	0.6500
single	transaction weight	$o_4, k=10$	0.7467
single	item weight	$o_4, k=10$	0.3300

Table C-3: Clustering metrics for transaction and item weighting, "Wavelets (1-500)" data set. Red and black text marks standard-vs-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	1.0000
complete	transaction weight	$o_3, k=1$	1.0000
complete	item weight	$o_3, k=1$	1.0000
average	pairwise	$o_3, k=1$	1.0000
average	transaction weight	$o_3, k=1$	1.0000
average	item weight	$o_3, k=1$	1.0000
single	pairwise	$o_3, k=1$	1.0000
single	transaction weight	$o_3, k=1$	1.0000
single	item weight	$o_3, k=1$	0.2143
complete	pairwise	$o_3, k=5$	0.5722
complete	transaction weight	$o_3, k=5$	0.5722
complete	item weight	$o_3, k=5$	0.5722
average	pairwise	$o_3, k=5$	0.5972
average	transaction weight	$o_3, k=5$	0.6333
average	item weight	$o_3, k=5$	0.3281
single	pairwise	$o_3, k=5$	0.4069
single	transaction weight	$o_3, k=5$	0.3622
single	item weight	$o_3, k=5$	0.1767
complete	pairwise	$o_3, k=10$	0.5222
complete	transaction weight	$o_3, k=10$	0.4528
complete	item weight	$o_3, k=10$	0.4528
average	pairwise	$o_3, k=10$	0.5336
average	transaction weight	$o_3, k=10$	0.5059
average	item weight	$o_3, k=10$	0.2222
single	pairwise	$o_3, k=10$	0.3609
single	transaction weight	$o_3, k=10$	0.2628
single	item weight	$o_3, k=10$	0.1403
complete	pairwise	$o_4, k=1$	1.0000
complete	transaction weight	$o_4, k=1$	1.0000
complete	item weight	$o_4, k=1$	1.0000

average	pairwise	$o_4, k=1$	1.0000
average	transaction weight	$o_4, k=1$	1.0000
average	item weight	$o_4, k=1$	0.1379
single	pairwise	$o_4, k=1$	0.2857
single	transaction weight	$o_4, k=1$	0.2353
single	item weight	$o_4, k=1$	0.1053
complete	pairwise	$o_4, k=5$	0.2593
complete	transaction weight	$o_4, k=5$	0.2593
complete	item weight	$o_4, k=5$	0.2593
average	pairwise	$o_4, k=5$	0.3296
average	transaction weight	$o_4, k=5$	0.4000
average	item weight	$o_4, k=5$	0.1459
single	pairwise	$o_4, k=5$	0.2684
single	transaction weight	$o_4, k=5$	0.2253
single	item weight	$o_4, k=5$	0.1290
complete	pairwise	$o_4, k=10$	0.1667
complete	transaction weight	$o_4, k=10$	0.2259
complete	item weight	$o_4, k=10$	0.1667
average	pairwise	$o_4, k=10$	0.2407
average	transaction weight	$o_4, k=10$	0.3556
average	item weight	$o_4, k=10$	0.1289
single	pairwise	$o_4, k=10$	0.2277
single	transaction weight	$o_4, k=10$	0.2034
single	item weight	$o_4, k=10$	0.1280

Table C-4: Clustering metrics for transaction and item weighting, "Wavelets and Brownian" data set. Red and black text marks standard-versus-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	1.0000
complete	transaction weight	$o_3, k=1$	1.0000
complete	item weight	$o_3, k=1$	0.5000
average	pairwise	$o_3, k=1$	1.0000
average	transaction weight	$o_3, k=1$	0.6000
average	item weight	$o_3, k=1$	0.5000
single	pairwise	$o_3, k=1$	0.6000
single	transaction weight	$o_3, k=1$	0.6000
single	item weight	$o_3, k=1$	0.4286
complete	pairwise	$o_3, k=5$	0.5600
average	transaction weight	$o_3, k=5$	0.6145
average	item weight	$o_3, k=5$	0.5857
average	pairwise	$o_3, k=5$	0.6857
average	transaction weight	$o_3, k=5$	0.6757
average	item weight	$o_3, k=5$	0.5857
single	pairwise	$o_3, k=5$	0.5657
single	transaction weight	$o_3, k=5$	0.6257
single	item weight	$o_3, k=5$	0.5714
complete	pairwise	$o_3, k=10$	0.5800
complete	transaction weight	$o_3, k=10$	0.5773
complete	item weight	$o_3, k=10$	0.6129
average	pairwise	$o_3, k=10$	0.6479
average	transaction weight	$o_3, k=10$	0.6229
average	item weight	$o_3, k=10$	0.6129
single	pairwise	$o_3, k=10$	0.5314
single	transaction weight	$o_3, k=10$	0.6229
single	item weight	$o_3, k=10$	0.5914
complete	pairwise	$o_4, k=1$	0.6667
complete	transaction weight	$o_4, k=1$	0.8000
complete	item weight	$o_4, k=1$	0.6667

average	pairwise	$o_4, k=1$	1.0000
average	transaction weight	$o_4, k=1$	0.8000
average	item weight	$o_4, k=1$	0.6667
single	pairwise	$o_4, k=1$	0.8000
single	transaction weight	$o_4, k=1$	0.8000
single	item weight	$o_4, k=1$	0.5714
complete	pairwise	$o_4, k=5$	0.6667
complete	transaction weight	$o_4, k=5$	0.7733
complete	item weight	$o_4, k=5$	0.7333
average	pairwise	$o_4, k=5$	0.7600
average	transaction weight	$o_4, k=5$	0.8133
average	item weight	$o_4, k=5$	0.7333
single	pairwise	$o_4, k=5$	0.7543
single	transaction weight	$o_4, k=5$	0.8133
single	item weight	$o_4, k=5$	0.6571
complete	pairwise	$o_4, k=10$	0.6810
complete	transaction weight	$o_4, k=10$	0.7143
complete	item weight	$o_4, k=10$	0.6689
average	pairwise	$o_4, k=10$	0.6933
average	transaction weight	$o_4, k=10$	0.7200
average	item weight	$o_4, k=10$	0.6467
single	pairwise	$o_4, k=10$	0.6514
single	transaction weight	$o_4, k=10$	0.6621
single	item weight	$o_4, k=10$	0.5956

Table C-5: Clustering metrics with bibliographic coupling for transaction and item weighting, "Wavelets and Brownian" data set. Red and black text marks standard-vs-hybrid distance comparisons.

Linkage	Distances	Frequent itemsets	Clustering metric
complete	pairwise	$o_3, k=1$	0.0508
complete	transaction weight	$o_3, k=1$	0.0508
complete	item weight	$o_3, k=1$	1.0000
average	pairwise	$o_3, k=1$	0.2500
average	transaction weight	$o_3, k=1$	0.6000
average	item weight	$o_3, k=1$	1.0000
single	pairwise	$o_3, k=1$	0.2308
single	transaction weight	$o_3, k=1$	0.6000
single	item weight	$o_3, k=1$	0.4286
complete	pairwise	$o_3, k=5$	0.3305
complete	transaction weight	$o_3, k=5$	0.3903
complete	item weight	$o_3, k=5$	0.4305
average	pairwise	$o_3, k=5$	0.5500
average	transaction weight	$o_3, k=5$	0.6900
average	item weight	$o_3, k=5$	0.6286
single	pairwise	$o_3, k=5$	0.5923
single	transaction weight	$o_3, k=5$	0.6900
single	item weight	$o_3, k=5$	0.2819
complete	pairwise	$o_3, k=10$	0.1907
complete	transaction weight	$o_3, k=10$	0.4853
complete	item weight	$o_3, k=10$	0.2407
average	pairwise	$o_3, k=10$	0.3125
average	transaction weight	$o_3, k=10$	0.6575
average	item weight	$o_3, k=10$	0.4015
single	pairwise	$o_3, k=10$	0.4115
single	transaction weight	$o_3, k=10$	0.6575
single	item weight	$o_3, k=10$	0.3221
complete	pairwise	$o_4, k=1$	0.0678
complete	transaction weight	$o_4, k=1$	1.0000
complete	item weight	$o_4, k=1$	0.0678

average	pairwise	$o_4, k=1$	0.1000
average	transaction weight	$o_4, k=1$	1.0000
average	item weight	$o_4, k=1$	0.2500
single	pairwise	$o_4, k=1$	0.3077
single	transaction weight	$o_4, k=1$	1.0000
single	item weight	$o_4, k=1$	0.4000
complete	pairwise	$o_4, k=5$	0.0678
complete	transaction weight	$o_4, k=5$	0.4007
complete	item weight	$o_4, k=5$	0.0678
average	pairwise	$o_4, k=5$	0.2400
average	transaction weight	$o_4, k=5$	0.6933
average	item weight	$o_4, k=5$	0.3103
single	pairwise	$o_4, k=5$	0.3077
single	transaction weight	$o_4, k=5$	0.7219
single	item weight	$o_4, k=5$	0.4015
complete	pairwise	$o_4, k=10$	0.0678
complete	transaction weight	$o_4, k=10$	0.3807
complete	item weight	$o_4, k=10$	0.0678
average	pairwise	$o_4, k=10$	0.1700
average	transaction weight	$o_4, k=10$	0.6900
average	item weight	$o_4, k=10$	0.2655
single	pairwise	$o_4, k=10$	0.3077
single	transaction weight	$o_4, k=10$	0.7114
single	item weight	$o_4, k=10$	0.3995

Appendix D

Minimum Spanning Tree Itemset-Connectedness Metrics

This appendix gives detailed results for the experiments conducted in Chapter 5, Section 5.4. In particular, it gives clustering metrics for each test case, along with the corresponding choice of experimental inputs. The results are presented Tables D-1 through D-10, corresponding to the 10 data sets employed.

Table D-1: Minimum spanning tree itemset-connectedness metrics for "Adaptive Optics" data set.

Distances	Frequent itemsets	MST metric
pairwise	$o_3, k=1$	1.0000
o_3^4	$o_3, k=1$	1.0000
o_3^6	$o_3, k=1$	1.0000
pairwise	$o_3, k=5$	0.8333
o_3^4	$o_3, k=5$	1.0000
o_3^6	$o_3, k=5$	1.0000
pairwise	$o_3, k=10$	0.4762
o_3^4	$o_3, k=10$	0.5000
o_3^6	$o_3, k=10$	0.5000
pairwise	$o_4, k=1$	0.2500
o_4^4	$o_4, k=1$	0.2500
o_4^6	$o_4, k=1$	0.2500
pairwise	$o_4, k=5$	0.2500
o_4^4	$o_4, k=5$	0.2500

o_4^6	$o_4, k = 5$	0.2500
pairwise	$o_4, k = 10$	0.2500
o_4^4	$o_4, k = 10$	0.2500
o_4^6	$o_4, k = 10$	0.2500

Table D-2: Minimum spanning tree itemset-connectedness metrics with bibliographic coupling for "Adaptive Optics" data set.

Distances	Frequent itemsets	MST metric
pairwise	$o_3, k=1$	0.5000
o_3^4	$o_3, k=1$	1.0000
pairwise	$o_3, k=5$	0.7143
o_3^4	$o_3, k=5$	0.7143
pairwise	$o_3, k=10$	0.6667
o_3^4	$o_3, k=10$	0.6250
pairwise	$o_4, k=1$	1.0000
o_4^4	$o_4, k=1$	1.0000
o_4^6	$o_4, k=1$	1.0000
pairwise	$o_4, k=5$	0.5000
o_4^4	$o_4, k=5$	0.4545
o_4^6	$o_4, k=5$	0.4545
pairwise	$o_4, k=10$	0.4167
o_4^4	$o_4, k=10$	0.4545
o_4^6	$o_4, k=10$	0.4545

Table D-3: Minimum spanning tree itemset-connectedness metrics for "Collagen" data set.

Distances	Frequent itemsets	MST metric
pairwise	$o_3, k=1$	1.0000
o_3^4	$o_3, k=1$	1.0000
o_3^4, min-sup = 2	$o_3, k=1$	1.0000
o_3^4, min-sup = 4	$o_3, k=1$	1.0000
pairwise	$o_3, k=5$	0.7143
o_3^4	$o_3, k=5$	0.7143
o_3^4, min-sup = 2	$o_3, k=5$	0.7143
o_3^4, min-sup = 4	$o_3, k=5$	0.7143
pairwise	$o_3, k=10$	0.7143
o_3^4	$o_3, k=10$	0.7143
o_3^4, min-sup = 2	$o_3, k=10$	0.7143
o_3^4, min-sup = 4	$o_3, k=10$	0.7143
pairwise	$o_4, k=1$	1.0000
o_4^4	$o_4, k=1$	1.0000
o_4^4, min-sup = 2	$o_4, k=1$	1.0000
o_4^4, min-sup = 4	$o_4, k=1$	1.0000
pairwise	$o_4, k=5$	0.5556
o_4^4	$o_4, k=5$	0.5556
o_4^4, min-sup = 2	$o_4, k=5$	0.4545
o_4^4, min-sup = 4	$o_4, k=5$	0.4545
pairwise	$o_4, k=10$	0.5556
o_4^4	$o_4, k=10$	0.5556
o_4^4, min-sup = 2	$o_4, k=10$	0.4762
o_4^4, min-sup = 4	$o_4, k=10$	0.4762

Table D-4: Minimum spanning tree itemset-connectedness metrics for "Genetic Algorithms and Neural Networks" data set.

Distances	Frequent itemsets	MST metric
pairwise	$o_3, k=1$	1.0000
o_3^4	$o_3, k=1$	1.0000
o_3^6	$o_3, k=1$	1.0000
pairwise	$o_3, k=5$	0.3846
o_3^4	$o_3, k=5$	0.3846
o_3^6	$o_3, k=5$	0.3846
pairwise	$o_3, k=10$	0.3571
o_3^4	$o_3, k=10$	0.3571
o_3^6	$o_3, k=10$	0.3571
pairwise	$o_4, k=1$	0.2500
o_4^4	$o_4, k=1$	0.2500
pairwise	$o_4, k=5$	0.2500
o_4^4	$o_4, k=5$	0.2500
pairwise	$o_4, k=10$	0.3846
o_4^4	$o_4, k=10$	0.4000

Table D-5: Minimum spanning tree itemset-connectedness metrics for "Quantum Gravity and Strings" data set.

Distances	Frequent itemsets	MST metric
pairwise	$o_3, k=1$	1.0000
o_3^4	$o_3, k=1$	1.0000
o_3^4, min-sup = 2	$o_3, k=1$	1.0000
o_3^4, min-sup = 4	$o_3, k=1$	1.0000
pairwise	$o_3, k=5$	1.0000
o_3^4	$o_3, k=5$	1.0000
o_3^4, min-sup = 2	$o_3, k=5$	1.0000
o_3^4, min-sup = 4	$o_3, k=5$	1.0000
pairwise	$o_3, k=10$	0.7692
o_3^4	$o_3, k=10$	0.7692
o_3^4, min-sup = 2	$o_3, k=10$	0.7692
o_3^4, min-sup = 4	$o_3, k=10$	0.7692
pairwise	$o_4, k=1$	1.0000
o_4^4	$o_4, k=1$	1.0000
o_4^4, min-sup = 2	$o_4, k=1$	1.0000
o_4^4, min-sup = 4	$o_4, k=1$	1.0000
pairwise	$o_4, k=5$	0.7143
o_4^4	$o_4, k=5$	0.7143
o_4^4, min-sup = 2	$o_4, k=5$	0.7143
o_4^4, min-sup = 4	$o_4, k=5$	0.7143
pairwise	$o_4, k=10$	0.4545
o_4^4	$o_4, k=10$	0.4545
o_4^4, min-sup = 2	$o_4, k=10$	0.4545
o_4^4, min-sup = 4	$o_4, k=10$	0.4545

Table D-6: Minimum spanning tree itemset-connectedness metrics with bibliographic coupling for "Quantum Gravity and Strings" data set.

Distances	Frequent itemsets	MST metric
pairwise	$o_3, k=1$	1.0000
o_3^4	$o_3, k=1$	1.0000
pairwise	$o_3, k=5$	1.0000
o_3^4	$o_3, k=5$	1.0000
pairwise	$o_3, k=10$	0.7692
o_3^4	$o_3, k=10$	0.7692
pairwise	$o_4, k=1$	1.0000
o_4^4	$o_4, k=1$	1.0000
o_4^6	$o_4, k=1$	1.0000
pairwise	$o_4, k=5$	1.0000
o_4^4	$o_4, k=5$	1.0000
o_4^6	$o_4, k=5$	1.0000
pairwise	$o_4, k=10$	0.9091
o_4^4	$o_4, k=10$	0.8333
o_4^6	$o_4, k=10$	0.9091

Table D-7: Minimum spanning tree itemset-connectedness metrics for "Wavelets (1-100)" data set.

Distances	Frequent itemsets	MST metric
pairwise	$o_3, k=1$	1.0000
o_3^4	$o_3, k=1$	1.0000
pairwise	$o_3, k=5$	0.7143
o_3^4	$o_3, k=5$	0.7143
pairwise	$o_3, k=10$	0.5556
o_3^4	$o_3, k=10$	0.7143
pairwise	$o_4, k=1$	1.0000
o_4^4	$o_4, k=1$	1.0000
pairwise	$o_4, k=5$	0.5556
o_4^4	$o_4, k=5$	1.0000
pairwise	$o_4, k=10$	0.6250
o_4^4	$o_4, k=10$	0.7143

Table D-8: Minimum spanning tree itemset-connectedness metrics for "Wavelets (1-500)" data set.

Distances	Frequent itemsets	MST metric
pairwise	$o_3, k=1$	1.0000
o_3^4	$o_3, k=1$	1.0000
o_3^4, min-sup = 2	$o_3, k=1$	1.0000
o_3^4, min-sup = 4	$o_3, k=1$	1.0000
pairwise	$o_3, k=5$	0.8333
o_3^4	$o_3, k=5$	0.8333
o_3^4, min-sup = 2	$o_3, k=5$	0.8333
o_3^4, min-sup = 4	$o_3, k=5$	1.0000
pairwise	$o_3, k=10$	0.7143
o_3^4	$o_3, k=10$	0.7143
o_3^4, min-sup = 2	$o_3, k=10$	0.7143
o_3^4, min-sup = 4	$o_3, k=10$	0.7143
pairwise	$o_4, k=1$	1.0000
o_4^4	$o_4, k=1$	1.0000
o_4^4, min-sup = 2	$o_4, k=1$	1.0000
o_4^4, min-sup = 4	$o_4, k=1$	1.0000
pairwise	$o_4, k=5$	0.7143
o_4^4	$o_4, k=5$	0.7143
o_4^4, min-sup = 2	$o_4, k=5$	0.7143
o_4^4, min-sup = 4	$o_4, k=5$	0.6250
pairwise	$o_4, k=10$	0.4000
o_4^4	$o_4, k=10$	0.4000
o_4^4, min-sup = 2	$o_4, k=10$	0.4000
o_4^4, min-sup = 4	$o_4, k=10$	0.4000

Table D-9: Minimum spanning tree itemset-connectedness metrics for "Wavelets and Brownian" data set.

Distances	Frequent itemsets	MST metric
pairwise	$o_3, k=1$	1.0000
o_3^4	$o_3, k=1$	1.0000
o_3^4, min-sup = 2	$o_3, k=1$	1.0000
o_3^4, min-sup = 4	$o_3, k=1$	1.0000
o_3^4, min-sup = 8	$o_3, k=1$	1.0000
pairwise	$o_3, k=5$	1.0000
o_3^4	$o_3, k=5$	1.0000
o_3^4, min-sup = 2	$o_3, k=5$	1.0000
o_3^4, min-sup = 4	$o_3, k=5$	1.0000
o_3^4, min-sup = 8	$o_3, k=5$	1.0000
pairwise	$o_3, k=10$	0.8333
o_3^4	$o_3, k=10$	0.8333
o_3^4, min-sup = 2	$o_3, k=10$	0.8333
o_3^4, min-sup = 4	$o_3, k=10$	0.8333
o_3^4, min-sup = 8	$o_3, k=10$	0.8333
pairwise	$o_4, k=1$	1.0000
o_4^4	$o_4, k=1$	1.0000
o_4^4, min-sup = 2	$o_4, k=1$	1.0000
o_4^4, min-sup = 4	$o_4, k=1$	1.0000
pairwise	$o_4, k=5$	1.0000
o_4^4	$o_4, k=5$	1.0000
o_4^4, min-sup = 2	$o_4, k=5$	1.0000
o_4^4, min-sup = 4	$o_4, k=5$	1.0000
pairwise	$o_4, k=10$	0.7692
o_4^4	$o_4, k=10$	0.7692
o_4^4, min-sup = 2	$o_4, k=10$	0.7692
o_4^4, min-sup = 4	$o_4, k=10$	0.7692

Table D-10: Minimum spanning tree itemset-connectedness metrics with bibliographic coupling for "Wavelets and Brownian" data set.

Distances	Frequent itemsets	MST metric
pairwise	$o_3, k=1$	0.5000
o_3^4	$o_3, k=1$	1.0000
o_3^4, min-sup = 2	$o_3, k=1$	1.0000
o_3^4, min-sup = 4	$o_3, k=1$	1.0000
pairwise	$o_3, k=5$	0.7143
o_3^4	$o_3, k=5$	1.0000
o_3^4, min-sup = 2	$o_3, k=5$	1.0000
o_3^4, min-sup = 4	$o_3, k=5$	1.0000
pairwise	$o_3, k=10$	0.7692
o_3^4	$o_3, k=10$	0.7692
o_3^4, min-sup = 2	$o_3, k=10$	0.7692
o_3^4, min-sup = 4	$o_3, k=10$	0.7692
pairwise	$o_4, k=1$	1.0000
o_4^4	$o_4, k=1$	1.0000
o_4^4, min-sup = 2	$o_4, k=1$	1.0000
o_4^4, min-sup = 4	$o_4, k=1$	1.0000
pairwise	$o_4, k=5$	0.5556
o_4^4	$o_4, k=5$	1.0000
o_4^4, min-sup = 2	$o_4, k=5$	1.0000
o_4^4, min-sup = 4	$o_4, k=5$	1.0000
pairwise	$o_4, k=10$	0.7143
o_4^4	$o_4, k=10$	1.0000
o_4^4, min-sup = 2	$o_4, k=10$	1.0000
o_4^4, min-sup = 4	$o_4, k=10$	1.0000

Abstract

This dissertation introduces new methods for visualizing collections of linked documents, for enhancing the understanding of relationships among documents that are returned by information retrieval systems. The methodology employs a new class of inter-document distances that capture the information inherent in the link structure of a collection. In particular, the distances are computed through the process of association mining, which results in the identification of sets of items (called itemsets) that are jointly linked to sufficiently often. In the context that links are citations appearing in published literature, itemsets are interpreted as higher-order co-citations. The new distances retain a simple pairwise structure, and are consistent with important frequently occurring itemsets. This approach provides the advantage that standard tools of visualization, e.g. hierarchical clustering and the minimum spanning tree can still be applied, while the distance information upon which they are based is richer. This work also proposes a number of enhancements to the standard visualizations, which support information retrieval tasks. The approach is demonstrated with document sets extracted from the Science Citation Index citation database. More generally, this work is applicable to information spaces in which objects may be associated by reference, e.g. software engineering, communications networks, and the World Wide Web.

Biographical Sketch

Steven E. Noel received his Bachelor of Science in Electro-Optics *Cum Laude* from the University of Houston – Clear Lake in 1989. He then began working as a physicist at the Naval Surface Warfare Center in Dahlgren, Virginia. As a Navy scientist he worked in digital signal processing, wavelets, neural networks, radar, infrared sensors, genetic algorithms, computer graphics, statistics, missile guidance, and astronomy. In 1998, he left the Navy in order to pursue an advanced degree in Computer Science from the University of Louisiana at Lafayette. He has published numerous refereed conference papers, Navy technical reports, and a paper in the *Journal of Electronic Imaging*. He co-chaired the Radar Signal Processing Session of the 1998 Wavelet Applications Conference, and served on the Tutorials Committee of the 1999 International Joint Conference on Neural Networks. He is a member of the Society of Photo-Optical Instrumentation Engineers, the International Neural Network Society, and the Alpha Chi, Phi Kappa Phi, and Upsilon Pi Epsilon honor societies.