Spectral Clustering for Microsoft Netscan Data
Technical Report UW-CSE-2005-06-05

Anne Patrikainen* Marina Meilă†

*Department of Computer Science and Engineering, University of Washington, Seattle. annep@cs.washington.edu.

†Department of Statistics, University of Washington, Seattle. mmp@stat.washington.edu.

Contents

Abstract

We present the results of exploratory data analysis for a data set that consists of crossposting information for 89,687 newsgroups over a period of 3.4 years. The data set we use is a part of Microsoft Netscan data. Our goal is to investigate the community structure of the newsgroup data set with a specific focus on spectral hierarchical clustering. We present a spectral hierarchical clustering algorithm and discuss existing and novel ways to measure the quality of a hierarchical clustering. We construct spectral hierarchical clusterings for ten subsets of the data set and evaluate the stability of the results.

keywords: Netscan project, Usenet, newsgroup, social network, spectral clustering, hierarchical clustering, natural community

1 Introduction

In this report, we present the results of exploratory data analysis on a part of the Microsoft Netscan data set. The Microsoft Netscan data contains information on the Usenet newsgroups from September 1999 to the present day. Usenet is a world-wide distributed discussion system. It consists of a set of "newsgroups" with names that are classified hierarchically by subject. "Messages" are "posted" to these newsgroups by people on computers with the appropriate software — these articles are then broadcast to other interconnected computer systems. [2]

Created in 1979 at the University of North Carolina, the Usenet initially connected only two computers, had 15 newsgroups, and handled a few messages per day [13]. In 1999, the Usenet was the third most widely used form of interaction media on the Internet in terms of users — the two leading ones were e-mail and the Web. At that time, the Usenet contained more than 14,000 newsgroups carrying 6 gigabytes of messages per day. On an average day, 20,000 people posted 300,000 messages [13]. The current number of Usenet newsgroups is more than 100,000. It is impossible to give a precise figure, since new groups are being born and old groups are dying every day, and further, not all groups are available everywhere in the world.

The form of social organization in the Usenet is unique. The newsgroups and the postings are stored in so-called news servers, which are located in all corners of the world. Anybody can turn her computer into a new news server, and it is totally up to the administrator which newsgroups are stored in the server. There is no single server that contains all newsgroups and all posts. Anybody can start a new newsgroup, but not all administrators of the news servers are willing to store the new group. Anybody can post a message to any newsgroup unless the group is moderated, in which case the newsgroup moderator has to accept the message before it is published. However, most newsgroups are not moderated, and it is possible to post any kinds of messages. The Usenet has no central authority, and no one owns most newsgroups; they are fully anarchic. Despite this, many newsgroups are well organized and productive. [13]

A *community* is a set of similar newsgroups. In this report, we investigate the community structure of the Usenet. Our focus is on spectral clustering of the Usenet newsgroups. A *clustering* is a grouping of data points into clusters such that the data points within a cluster are close to each other but far from the data points in the other clusters. A *clustering algorithm* is an algorithm whose output is a clustering, and *spectral clustering* is a class of clustering algorithms. Spectral clustering algorithms take as an input the pairwise similarities of the data points (in this case newsgroups), and employ the eigenvectors and eigenvalues of the similarity matrix or related matrix in clustering the data. Spectral clustering algorithms have been successfully used for instance in image segmentation. [10, 12, 18, 19, 17, 9]

Previously, spectral clustering has been applied to newsgroup data in [3]. We are not aware of any other articles that study clustering of newsgroups. However, the community structure of the World Wide Web has received more attention [7, 6, 4, 16]. Since both Usenet and the Web are essentially social networks, relatively similar methods can be applied to both.

The report is organized as follows. We present a brief review of spectral clustering and related concepts in Chapter 2. We give an overview of our newsgroup data set in Chapter 3. In Chapter 4, we introduce a method that we use to compute spectral hierarchical clusterings for newsgroup data. Chapter 5 is devoted to discussing how we can measure the quality of a hierarchical clustering. In Chapter 6, we discuss a method for identifying natural communities in the Usenet data. To demonstrate these methods and algorithms, we present a detailed case study of a subset of our data set, the so-called `talk` data, in Chapter 7. In Chapter 8, we present similar results for nine additional subsets of the newsgroup data. Finally, we conclude in Chapter 9.

2 Overview of Spectral Clustering

Let us consider a set V of N data points, or vertices. Let us write S_{ij} for the similarity between the ith and the jth data point, and $S = (S_{ij})$ for the $N \times N$ *similarity matrix*. In the following, we will only consider symmetric similarity matrices. Let us define the *volume* D_i of vertex $i \in V$ by

$$D_i = \text{Vol } i = \sum_{j \in V} S_{ij}.$$

Without loss of generality, we assume that all vertices have non-zero volumes.

Next, let us write $D = (D_{ij})$ for the $N \times N$ diagonal matrix with $D_{ii} = D_i$. Finally, we define the volume of a set A of vertices as

$$\text{Vol } A = \sum_{i \in A} D_i.$$

A useful way to analyze spectral clustering is to consider it in terms of a random walk on the set of vertices. To this end, we form the stochastic

transition matrix $P = (P_{ij})$ by normalizing the row sums of S to 1. Formally,

$$P = D^{-1}S, \tag{1}$$

or equivalently, $P_{ij} = S_{ij}/D_i$. We can interpret P_{ij} as the *transition probability* $\mathrm{P}(i \to j|i)$ of moving from vertex i to vertex j, given that the random walk starts from vertex i. The eigenvalues of P are $1 = \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq -1$ and the corresponding eigenvectors are $v^1, v^2, \ldots, v^n$. The eigenvalues of P are real and the eigenvectors linearly independent. If the *eigengap* $\Delta_k = \lambda_k - \lambda_{k+1}$ is large, the subspace spanned by the eigenvectors $v^1, v^2, \ldots, v^k$ is stable to perturbations.

A *stationary distribution* $\pi = (\pi_i)_{i \in V}$ of a Markov chain is a probability distribution over the vertices in V such that $P^T\pi = \pi$. In our setting, the stationary distribution values are given by

$$\pi_i = \frac{D_i}{\mathrm{Vol}\ V}.$$

If we have a subset $A \subseteq V$, let us write $\pi_A = \mathrm{Vol}\ A/\mathrm{Vol}\ V$ for the probability of A under the stationary distribution.

A *clustering* $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$ is a partition of V into disjoint non-empty sets $C_1, C_2, \ldots, C_K$. The quality of the clustering depends on the similarities of the vertices within each cluster, on the dissimilarities of the vertices in different clusters, and on the sizes of the clusters. There are number of ways to formalize the quality of a clustering. One of the most widely used clustering criteria is *multiway normalized cut*, given as

$$\mathrm{MNCut}(\mathcal{C}) = \sum_{k=1}^{K} \sum_{k' \neq k} \frac{\mathrm{Cut}(C_k, C_{k'})}{\mathrm{Vol}\ C_k},$$

where

$$\mathrm{Cut}(A, B) = \sum_{i \in A} \sum_{j \in B} S_{ij}.$$

Small values of the criterion are desirable.

In the special case of two clusters, the multiway normalized cut reduces into

$$\mathrm{MNCut}(\{C_1, C_2\}) = \frac{\mathrm{Cut}(C_1, C_2)}{\mathrm{Vol}\ C_1} + \frac{\mathrm{Cut}(C_1, C_2)}{\mathrm{Vol}\ C_2}.$$

By inspecting this expression, it is easy to see that a small value is achieved when the clusters have balanced volumes and when $\mathrm{Cut}(C_1, C_2)$ is small, or in other words, the vertices in the two clusters are very dissimilar. If we have a bipartite graph which we partition into two sets C_1 and C_2 such that all edges are included in the cut, we obtain the maximum normalized cut value 2, since $\mathrm{Cut}(C_1, C_2) = \mathrm{Vol}\ C_1 = \mathrm{Vol}\ C_2$. The minimum normalized cut value 0 is achieved when C_1 are C_2 are disconnected, or in other words, $\mathrm{Cut}(C_1, C_2) = 0$.

The multiway normalized cut criterion has an intuitive interpretation in terms of the Markov random walk view. Given two vertex sets $A \subset V$ and

$B \subset V$, we define $P_{AB} = \mathrm{P}(A \to B|A)$ as the probability of the random walk going from the set A to the set B in one step given that the current state is in A and the random walk is in its stationary distribution π. We can write out this expression as

$$P_{AB} = \frac{\sum_{i \in A, j \in B} \pi_i P_{ij}}{\pi_A} = \frac{\sum_{i \in A, j \in B} S_{ij}}{\mathrm{Vol}\, A} = \frac{\mathrm{Cut}(A, B)}{\mathrm{Vol}\, A}. \tag{2}$$

We now see that the multiway normalized cut is in fact a sum of "out-of-cluster" transition probabilities:

$$\mathrm{MNCut}(\mathcal{C}) = \sum_{k=1}^{K} \sum_{k \neq k'} P_{C_k C_{k'}} = K - \sum_{k=1}^{K} P_{C_k C_k}.$$

It is now clear why small values of the MNCut criterion are desirable. This kinds of values are achieved for partitions in which the probability of the random walk leaving a cluster is small.

It has been shown in [9] that

$$\mathrm{MNCut}(\mathcal{C}) \geq K - \sum_{k=1}^{K} \lambda_k(P).$$

The difference between the MNCut value and its lower bound is referred to as the *gap*, given as

$$\mathrm{gap}_P(\mathcal{C}) = \mathrm{MNCut}(\mathcal{C}) - K + \sum_{k=1}^{K} \lambda_k(P).$$

The matrix P has *piecewise constant eigenvectors* $v^1, v^2, \ldots, v^K$ w.r.t. a clustering $\mathcal{C}$ if $v_i^k = v_j^k$ for all $k \leq K$ whenever the vertices i and j are in the same cluster. It has been shown that $\mathrm{gap}_P(\mathcal{C}) = 0$ if and only if P has piecewise constant eigenvectors w.r.t. $\mathcal{C}$. [9]

A *spectral clustering algorithm* is an algorithm that clusters the set of vertices V in a way that utilizes the eigenvalues and the eigenvectors of a matrix derived from the similarity matrix S. There are several different spectral clustering algorithms [10, 12, 18, 19, 17, 9]. Let us introduce an example of a spectral clustering algorithm as Algorithm 1 [9].

It can be shown that Algorithm 1, as well as many other spectral clustering algorithms, produce clusterings that minimize the multiway normalized cut criterion in certain special cases, as the following theorem indicates. [9]

Theorem 1 (Multicut lemma) *Let S be an $N \times N$ symmetric matrix with nonnegative elements, and let P be the corresponding transition probability matrix. Assume that P has K piecewise constant eigenvectors $v^1, v^2, \ldots, v^K$ w.r.t. a clustering $\mathcal{C}$, $|\mathcal{C}| = K$. Denote the corresponding eigenvalues by $\lambda_1, \lambda_2, \ldots, \lambda_K$ and assume that these are the K largest eigenvalues P, are all non-zero, and $\lambda_K \geq \lambda_{K+1}$. The the minimum K-way normalized cut for S is given by the partition $\mathcal{C}$.*

Algorithm 1. Spectral Clustering Algorithm.
Input: $N \times N$ Similarity matrix S, desired number of clusters K.
Output: Clustering $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$.

1. Compute the transition probability matrix P.
2. Compute $v^1, v^2, \ldots, v^K$, the eigenvectors corresponding to the K largest eigenvalues of P. Form a matrix $V = (v^1\ v^2\ \ldots\ v^K)$ whose columns are these eigenvectors.
3. Cluster the rows of V as points in $\mathbb{R}^{\mathbb{K}}$ into K clusters for instance with the K-means algorithm.

3 Overview of the Data

Our data set is a subset of the Microsoft Netscan Usenet data. The Netscan group at Microsoft Research has collected information on Usenet newsgroups from September 1999 to present day. A part of this data has been aggregated and made available to the researchers of the University of Washington.

The Usenet consists of *newsgroups* with names such as

`comp.os.linux.development.system`,
`soc.religion.christian.bible-study`,
`atl.sports.baseball.atlanta-braves`,
`alt.amazon-women.admirers`,
`rec.crafts.textiles.sewing`, or
`sci.geo.earthquake`.

The names are composed of hierarchical parts of increasing specificity. For instance, in `comp.os.linux.development.system`, `comp` stands for computing-related topics, `os` stands for operating systems, `linux` is a specific operating system, and so on. We refer to sets of newsgroups whose names begin with `comp`, `soc`, etc. as *first-level hierarchies*. Sets of newsgroups whose names begin with `comp.os`, `soc.religion`, etc. are referred to as *second-level hierarchies*. A monthly updated master list of first-level hierarchies and their descriptions can be found at `http://www.magma.ca/%7Eleisen/mlnh/mlnhtables.html`.

Each newsgroup contains *posts*, messages which are related to the topic of the newsgroup. Any post can be replied to: a chain of posts formed this way is referred to as a *thread*. A post can be replied to a number of times, so a post and all replies associated to it form a tree structure. A *start* is a post that has been replied to but is not a reply itself; a *barren* is a post that has not been replied to and is not a reply itself. A *crosspost* is a message that has been sent to multiple newsgroups.

Our data set contains information on crosspostings in the Usenet during the period January 2000 – May 2003. During this time, 711,857,644 crosspostings were recorded in 89,687 newsgroups. Any pair of newsgroups in our set had

Number of cross-posted messages	Hierarchy name	Topic
475246642	alt	Alternative
26327267	soc	Social issues
13517913	tw	Taiwan
13316980	rec	Recreation
11719741	talk	Talk
10012143	comp	Computing
9944991	uk	UK
9556241	clari	Clarinet News Service (commercial)
9112071	microsoft	Microsoft
8275892	relcom	Commonwealth of Ind. States (Cyrillic)
7017797	misc	Miscellaneous
4773233	sci	Science
4501469	aus	Australia
4253506	ukr	Ukraine
4235915	news	Usenet news
4177477	es	Spain
3301752	it	Italy
3211799	us	US
3179200	yu	Yugoslavia
3178924	hk	Hong Kong

Table 1: Twenty largest first-level newsgroup hierarchies based on the number of crosspostings.

thus an average of 0.18 shared messages. Note, however, that we do not have any information on the newsgroups with no crosspostings in them. Out of the 711,857,644 crosspostings, 447,860,777 were replies, 240,502,969 were starts, and 23,493,898 were barrens.

In our data set, there are 3,157 level 1 hierarchies, 23,319 level 2 hierarchies, and 59,642 level 3 hierarchies. Table 3 contains the 20 largest level 1 hierarchies ranked by the number of crosspostings (starts, barrens, and replies) in them. Table 3 contains the 20 largest level 2 hierarchies ranked by the same criterion. In Table 3, we have listed the 20 largest level 1 hierarchies with respect to the number of newsgroups in them. Note that a large number of crosspostings does not necessarily imply a large number of newsgroups, or vice versa.

Number of crossposted messages	Hierarchy name
253592506	alt.binaries
45827774	alt.sex
29757783	alt.politics
23531713	soc.culture
12937062	tw.bbs
11990596	alt.religion
11918677	alt.fan
9001759	microsoft.public
8892829	talk.politics
7721995	relcom.commerce
5825242	alt.bainaries
4337069	alt.christnet
3996712	ukr.commerce
3863846	alt.society
3511458	alt.music
3357161	clari.web
3258662	misc.jobs
3027809	alt.bestjobsusa
2868852	alt.games
2857253	alt.atheism

Table 2: Twenty largest second-level newsgroup hierarchies based on the number of crosspostings

Number of newsgroups	**Hierarchy name**	**Topic**
29287	`alt`	Alternative
3768	`free`	Entirely unregulated newsgroups
3245	`microsoft`	Microsoft
1596	`comp`	Computing
1285	`rec`	Recreation
1192	`fido7`	Russian-language Fidonet
1181	`clari`	Clarinet news service (commercial)
1040	`news`	Usenet news
960	`de`	Germany
835	`aol`	America Online (ISP)
804	`fido`	Fidonet
624	`it`	Italy
615	`uw`	University of Waterloo, Ontario, Canada
596	`fj`	Japan
575	`japan`	Japan
542	`uk`	UK
523	`z-netz`	Z-Netz (German newsgroups)
523	`tw`	Taiwan
522	`soc`	Social issues
467	`ucb`	University of California at Berkeley, USA

Table 3: Twenty largest first-level newsgroup hierarchies based on the number of newsgroups in the hierarchy

Component size	Number of components
88999	1
19	2
11	2
10	1
9	2
8	1
7	1
6	10
5	29
4	26
3	193
2	131

Table 4: Sizes and numbers of connected components in the newsgroup data based on all crosspostings.

3.1 Usenet as a Graph

A *graph* is defined as a pair (V, E), where V is a collection of *nodes* or *vertices*, and E is a collection of *edges* (vertex pairs). Let us consider the newsgroups as vertices of a graph. If a pair of newsgroups shares a crossposting or several crosspostings, there is an edge between the corresponding vertices, and the weight of the edge is the number of shared crosspostings. The newsgroup graph can be represented by a *newsgroup similarity matrix* $S = (S_{ij})$, which is formed by setting the similarity S_{ij} to equal the number of crosspostings between the ith and the jth newsgroup. Unless otherwise mentioned, we $S_{ii} = 0$ for all i.

A *degree* of a vertex is the sum of the weights of the edges incident to it. A degree of a newsgroup in a newsgroup graph is therefore the total number of crossposted messages in the group. Note that the degree of the ith newsgroup is equal to the volume D_i as defined in Section 2, and the diagonal matrix D has all newsgroup degrees on its diagonal.

A *path* is a sequence of consecutive edges in a graph. A graph is *connected* is there is a path connecting every pair of vertices. A graph that is not connected can be divided into *connected components* (disjoint connected subgraphs). Our newsgroup graph is not connected — Table 3.1 shows how many connected components it has and what are the sizes of those components. The size of the largest connected component is 88,999 newsgroups (vertices). The next largest component has only 19 newsgroups. Note that we do not have any components of size 1, since we have only included newsgroups which have at least one crossposting with another group.

A part of newsgroup postings are *spam* messages (unsolicited bulk postings that usually advertise a product). Spam messages are likely to be barrens, since they do not often get replied, nor are they replies themselves. It is possible that a given spam message gets sent to a large number of unrelated newsgroups,

Component size	Number of components
63115	1
12	1
11	2
10	1
9	2
8	4
7	4
6	8
5	8
4	28
3	71
2	90

Table 5: Sizes and numbers of connected components in the newsgroup data based on starts and replies only.

thereby reducing the number of connected components in the newsgroup graph. To alleviate the effect of spam, we eliminated all barrens and recomputed the connected components in the graph. The results are presented in Table 3.1. They look reasonably similar: we have one large connected component of size 63,115, and the second largest connected component has only 12 newsgroups. Note that the total number of newsgroups in Table 3.1 is less than the total number of newsgroups in Table 3.1, since we have only included groups with a non-zero number of start or reply crosspostings in the former.

4 Spectral Hierarchical Clustering of Newsgroups

We can compute a spectral clustering for Usenet newsgroups based on the crossposting matrix S for instance with Algorithm 1. However, it is not clear what value we should choose for the number of clusters K. Further, it is likely that a flat clustering cannot fully capture the complicated structure of the data set. For these reasons, we will attempt to construct a hierarchical clustering for the Usenet newsgroups with the help of spectral clustering algorithms.

Our approach consists of repeatedly splitting the set of vertices into two parts, until only singleton clusters remain. We would like to choose splits corresponding to small values of the 2-way normalized cut criterion (from now on, we refer to the normalized cut criterion simply as a 'cut'). To this end, the splitting is done with the help of the second eigenvector of the transition probability matrix P.

Let us now describe our approach in more detail. Consider a set V of N vertices and the $N \times N$ similarity matrix S. We would like to split the set of N vertices into two parts such that the cut is minimized. It is computationally intractable to try out all possible splits, so we use spectral clustering to choose

a split with a small cut value. Let us compute the transition probability matrix P and the eigenvector v^2 corresponding to the second largest eigenvalue of P. Let us sort the N vertices according to the values in the $N \times 1$ vector v^2. Let us then try splitting the N vertices into two sets in $N-1$ ways according to the order imposed by v^2. We compute the cut corresponding to each of the $N-1$ splits and choose the split with the smallest cut. We now have two sets of vertices V_1 and V_2 with respective sizes N_1 and N_2, where $N_1 + N_2 = N$. We proceed to split each of these vertex sets further, until only singleton clusters remain.

The result of a hierarchical clustering algorithm can be visualized as a *dendrogram*, a tree-like structure. From now on, we use the words hierarchical clustering and dendrogram interchangeably.

We would like to prune our hierarchy such that all 'bad' splits are undone. However, evaluating the goodness of a split is not trivial. We could choose a cut threshold c and remove all splits whose cut values exceed the threshold. We could do pruning also based on the gap value, or the combination of the cut and the gap value. Yet another alternative would be to prune based on cut/ $<$ cut $>$, where $<$ cut $>$ is the mean value of all possible cuts for the graph at hand. This choice might eliminate the possible effect of the graph size on the cut value. Similar normalization could be done for the gap value. Based on theoretical considerations and experimental results (see the discussion in Section 7.2), we have decided upon pruning based on the cut value alone. The appropriate cut threshold will be decided for each data set separately based on the cut-gap graph (again, see Section 7.2 for pruning examples).

Even after pruning the dendrogram according to the cut value, we do not have a fully reliable picture of the clustering structure of the data. Consider a subset V_s of the vertices which we have decided not to split further because of too big cut value. It is possible that the newsgroups in V_s form a strongly connected graph, in which case it is justified not to split the cluster further. On the other hand, the newsgroups in V_s might be very weakly connected but in a way that there is no obvious way to split the data into two parts, so the cut value will be large. In this case, the newsgroups in V_s should not be considered as a cluster — one possibility would be to split V_s into multiple small clusters, maybe even into singleton clusters. Deciding upon a good way to process a given cluster is highly nontrivial task and is left for future work.

As mentioned earlier, hierarchical clusterings are commonly visualized by means of a dendrogram. In the special case of spectral hierarchical clusterings, we can use a *dendromatrix* as an alternative, more expressive way to visualize the result. If we have a clustering on N vertices, the dendromatrix is a black-and-white matrix of size $N \times N$. The division of the vertices into clusters is represented by horizontal and vertical lines in appropriate locations. The colors in the matrix represent the transition probabilities between the clusters or between individual vertices.

There are two kinds of dendromatrices: *unit dendromatrices* and *block dendromatrices*. A unit dendromatrix is colored simply according to the $N \times N$ transition probability matrix P. The color of each unit dendromatrix element

depends on the corresponding transition probability P_{ij}. In a block dendromatrix, the color of a block (C_i, C_j) represents the probability $P_{C_iC_j}$ of the random walk going for the set of vertices C_i to the set of vertices C_j — the expression is given in Eq. 2. We will show examples of dendromatrices in Section 7.2.1.

5 Stability of Hierarchical Clustering

After computing a hierarchical clustering, it is important to evaluate the stability of the result. Would a small modification in the newsgroup graph cause a big change in the hierarchical structure, or is the structure robust to minor alterations?

Consider a hierarchical clustering $\mathcal{H}$ formed by applying spectral hierarchical clustering on a similarity matrix S. Let us modify S slightly, produce a new similarity matrix S', and compute a new hierarchical clustering $\mathcal{H}'$. Calculating the distance $d(\mathcal{H}, \mathcal{H}')$ between the two dendrograms will now give us information on the stability of the original clustering $\mathcal{H}$. If we repeat the experiment several times and get consistently small values for the distance $d(\mathcal{H}, \mathcal{H}')$, we may conclude that the clustering $\mathcal{H}$ is stable.

There are several important details of this process to consider, and each of them has a significant effect on the final stability result. Calculating the distance $d(\mathcal{H}, \mathcal{H}')$ is highly nontrivial — we will discuss different alternatives in Section 5.2. There are several ways to modify the similarity matrix S; the most intuitive ones are perhaps adding noise to it, and sampling data points (newsgroups) from it. Since it is not clear what would be a good way to add noise to a sparse similarity matrix, and since the noise level should be a function of the number of data points, we will not consider adding noise as an option at all. Instead, we will discuss various ways of sampling the set of data points in Section 5.1.

When we have computed hierarchical clustering $\mathcal{H}$ for a similarity matrix S, we have pruned the dendrogram according to the cut threshold c. After we form S' and compute $\mathcal{H}'$, we will have to have pruned the dendrogram accordingly. However, this is not straightforward, since the new cut threshold c' should depend on the sample percentage / noise level. We have simply set $c' = c$, but in the future research, the choice of c' should be given careful consideration.

5.1 Sampling the Data

Let us consider a similarity matrix S of size $N \times N$. We wish to sample the set of N data points (newsgroups) according to a sample percentage p. There are several different ways to sample the data: we introduce three possibilities here.

- In *uniform sampling*, we remove p percent of the data points at random. However, recall that many newsgroup dendrograms contain one big leaf cluster and several leaf clusters with only few newsgroups each. Removing a single point from a very small cluster is likely to cause significant change in that cluster, so we want to consider alternative ways of sampling.

- In *substructure sampling*, we choose one of the leaf clusters at random, with probability proportional to its size. We then remove all data points in this cluster. The biggest cluster is an exception: we never remove it. We repeat this procedure until we have less than $(1-p)N$ data points left. Since we are removing several data points at a time, the percentage of data points removed is rarely equal to p — it is often larger than that.

- In so-called *big cluster sampling*, we remove points at random only from the biggest leaf cluster, and leave all other leaf clusters untouched. If the size of the biggest cluster is less than pN, we simply remove all points in the biggest cluster and nothing else. The percentage of the points removed is therefore sometimes less than p.

5.2 Comparing Hierarchical Clusterings

There are several reasons for why we would like to compare hierarchical clusterings for newsgroup data; we list some examples below.

- We could evaluate the stability of a clustering by sampling the data, reclustering, and comparing the resulting clustering to the original one.

- To see how the clustering changes over time, we could cluster a few months' worth of data at a time, and compare the clusterings for different time slots. Alternatively, we could utilize sliding window over time.

- We could compare the hierarchical spectral clustering results with the newsgroup name hierarchy.

Unfortunately, not much research has been done on developing methods for comparing hierarchical clusterings. Maybe the simplest approach is to compare the leaf clusterings by calculating the value of the Clustering Error (CE), Variation of Information (VI), Rand index, or some other distance measure for ordinary clusterings (partitions of a set of data points) [8]. However, this approach does not take in account the cluster hierarchy in any way.

The only existing hierarchical clustering comparison methods that we are aware of are simple layer-by-layer comparison [5], utilizing so-called cophenetic matrices [15], or employing the subspace clustering CE measure [11]. These have several shortcomings: the layer-by-layer and the subspace clustering CE method are not able to take into account the cut values; and the layer-by-layer and the cophenetic matrix methods are not able to handle clusterings on different number of data points.

Let us now describe the cophenetic matrix comparison method in more detail. To this end, we define the *unweighted cophenetic distance* $d_{ij}^{uc}(\mathcal{D})$ between the ith and the jth newsgroup as the number of splits that separate them in the dendrogram $\mathcal{D}$.[1] The *weighted cophenetic distance* $d_{ij}^{wc}(\mathcal{D})$ between the ith and the jth newsgroup is the sum of the weights associated with the splits that

[1] Note the close similarity to the so-called tree distance.

separate the two newsgroups in the dendrogram. There are numerous ways to define the weights: for instance, we might choose $exp(-\text{Cut})$, since small cut values should correspond to large weight values (large distances between the clusters) and vice versa.

An *unweighted cophenetic matrix* $H^{uc} = (d_{ij}^{uc}(\mathcal{D}))$ consists of unweighted cophenetic distances, and a *weighted cophenetic matrix* $H^{wc} = (d_{ij}^{uc}(\mathcal{D}))$ consists of weighted cophenetic distances. Both types of matrices are always symmetric. A hierarchical clustering algorithm can be viewed as a mapping of the similarity matrix into a cophenetic matrix.[2]

Two hierarchical clusterings can be compared by comparing the corresponding cophenetic matrices. However, there are multiple ways to compare cophenetic matrices $H = (H_{ij})$ and $H' = (H'_{ij})$. Among other possibilities, we could compute the correlation of the matrix elements, the L1 distance between the matrix elements, or the L2 distance between the matrix elements. We are not aware of any research on the properties of different methods. We will therefore use several methods in our comparison experiments.

If we want to compare a cophenetic matrix of hierarchical clustering with a cophenetic matrix of a clustering on a sample of the data points, we have to scale the matrices appropriately with a function of the number of data points. Also, to make the stability measures comparable across different data sets, we would like the measure to be independent of the number of data points in the sample. Consider having a cophenetic matrix H for a hierarchical clustering $\mathcal{H}$ of N data points and a cophenetic matrix H' for a hierarchical clustering $\mathcal{H}'$ on a sample of $(1-p)N$ data points. That is, H is of size $N \times N$ and H' is of size $(1-p)N \times (1-p)N$.

In order to compare these matrices using correlation or L1/L2 distance, we have to extract the submatrix of size $(1-p)N \times (1-p)N$ from H; let us refer to this submatrix as H_{sub}. But H_{sub} is likely to have larger values in it on average than H', since $\mathcal{H}$ is a clustering on a bigger set of data points, and the corresponding dendrogram is often deeper. Unfortunately, it is not clear what is a good way to scale the cophenetic matrices. We could scale H and H_{sub} by $\log_2(N)$, the minimum depth of an unpruned tree, by N, the maximum depth of an unpruned tree, or something else. Alternatively, we might normalize the sum of the elements in the cophenetic matrix to 1 (L1 normalization), or the square root of the sum of the squared elements of the cophenetic matrix to 1 (L2 normalization). In the absence of theoretical and experimental results on different scaling factors, we will run experiments with cophenetic matrices that are scaled and normalized in various ways.

6 Natural Communities in Usenet

Let us refer to a cluster of similar newsgroups as a *community*. Running a clustering algorithm on the set of newsgroups can give us information on the

[2]Note that we have slightly modified the definitions of cophenetic distance and cophenetic matrix; the original definitions can be found in [15].

community structure of the Usenet. However, different clustering algorithms give different results, choices of parameter values might have a large effect too, and some clustering algorithms are nondeterministic. Since we cannot necessarily trust a single clustering result, we are interested in comparing and combining multiple clusterings. If a set of newsgroups is clustered together in a large number of clustering, we refer to this set as a *natural community*. Natural communities of World Wide Web pages have been studied in [6, 7], but we are not aware of any article on natural communities in newsgroup data.

Consensus clustering, also known as cluster ensemble or aggregate clustering, is a way to combine several clustering results into a single clustering [14]. We do not provide a literature survey here but discuss a specific application of consensus clustering, namely combining several K-means clusterings into a so-called mean connectivity matrix. The result of the K-means clustering algorithm depends on its initialization. We can combine the results of R runs of K-means into a *mean connectivity matrix* $M = (M_{ij})$ [14]. If we have N newsgroups, the mean connectivity matrix is of size $N \times N$. The entry M_{ij} represents the fraction of the R runs in which the ith and the jth newsgroup have been clustered together. Naturally, $0 \leq M_{ij} \leq 1$ for all i, j. If the value of M_{ij} is close to one, we have a good reason to believe that the corresponding two newsgroups are very closely related. Identifying blocks of large values in the mean connectivity matrix can help us reveal stable clusters, or natural communities, in the data.

An interesting avenue to explore in the future would be to extract the set of natural communities in various points of time, find correspondences between the communities across the time points, and investigate the temporal development of the community structure of the Usenet. In fact, this type of study has been conducted in [16] for Japanese Web pages. In this study, the researchers look in detail how Web communities emerge, dissolve, grow, shrink, are split into several communities, and are merged with another community.

7 Case Study: Talk Data

In this section, we will look at the `talk` newsgroup hierarchy in detail and apply all the methods presented in Sections 4, 5, and 6.

7.1 Overview of Talk Data

The first-level newsgroup hierarchy `talk` has 138 newsgroups and 920,858 crosspostings, out of which 832,942 are replies, 34,880 are starts, and 53,036 are barrens. We have 1 connected component of size 131 and 1 connected component of size 7. The groups in the smaller connected component are

`talk.hh.ii.pp`,
`talk.hh.ii.pp.cc.rr.ii.mm.ee.congressagent`,
`talk.hh.ii.pp.cc.rr.ii.mm.ee.activeagent`,
`talk.hh.ii.pp.cc.rr.ii.mm.ee`,
`talk.hh.ii.pp.cc.rr.ii`,

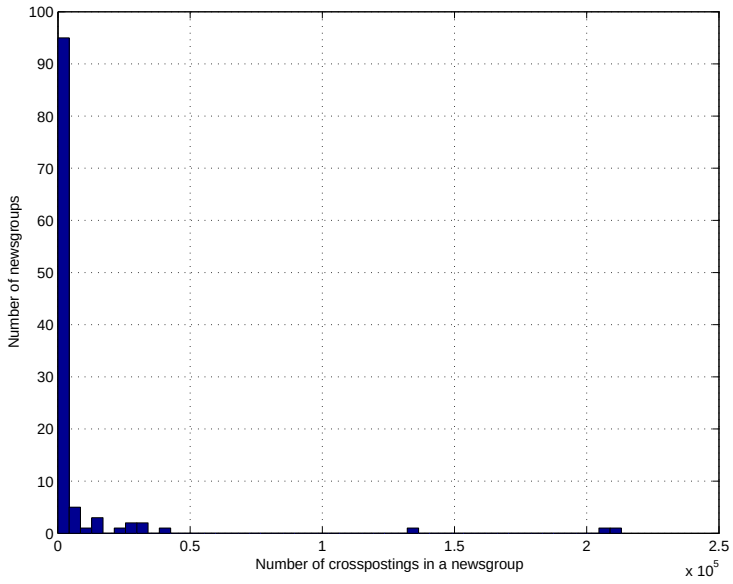


Figure 1: Number of crosspostings in the `talk` hierarchy.

`talk.hh`, and
`talk.hh.ii.pp.cc.rr.ii.mm`.[3]

To reduce the effect of spam messages, we leave all barrens out of considerations. After this, we have 113 newsgroups with a non-zero number of crosspostings. These newsgroups form a single connected component. The largest number of crosspostings in a newsgroup is 213,121 and the smallest number is 1. Fig. 7.1 shows the histogram for the numbers of crosspostings. Table 6 contains a list of all level 2 name hierarchies in the `talk` data.

[3]The alias HipCrime has written software to allow abusing Usenet in various ways, for instance flooding.

Number of newsgroups	Hierarchy name
53	talk.religion
31	talk.politics
7	talk.superphone
7	talk.hh
6	talk.philosophy
5	talk.hipcrime
5	talk.hipclone
5	talk.crimehip
4	talk.h
4	talk.emircpih
3	talk.bizarre
2	talk.h2pcr2me
2	talk.h1pcr1me

Table 6: Level 2 hierarchies in the talk data. In addition to the hierarchies listed above, there are 33 level 2 hierarchies with only one newsgroup.

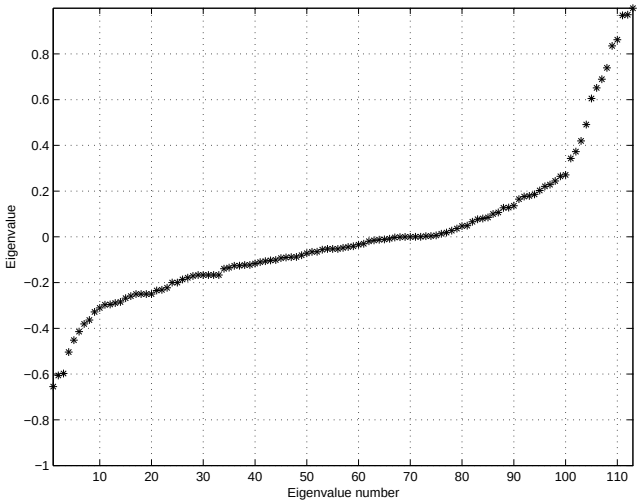


Figure 2: The eigenvalues of P for the `talk` hierarchy.

7.2 Spectral Hierarchical Clustering for Talk Data

We form the 113×113 similarity matrix S for the talk data by setting S_{ij} equal to the number of crosspostings (starts and replies) shared by the ith and the jth newsgroup. The diagonal entries of S are zero. We compute the stochastic transition matrix P by normalizing the rows of S according to Eq. 1. The eigenvalues of P are displayed in Fig. 2. As expected, the eigenvalues fall between -1 and 1. There are large gaps after the 3rd, the 5th, and the 9th largest eigenvalue. The eigenvector corresponding to the second largest eigenvector is shown in Fig. 3. The spikes correspond to newsgroups in the talk.religion.christian hierarchy. In fact, quite a few of the eigenvectors are spiky. Fig. 4 shows a plot of the second and the third eigenvector; most of the newsgroups remain at origin.

Let us construct a hierarchical clustering of the talk newsgroups by following the procedure described in Section 4. We repeatedly split the data matrix into two parts based on the value of the normalized cut. We continue splitting until no cluster contains more than one newsgroup.

To get a flavor of the magnitude of the cut and the gap values, we have plotted all these for the talk data in Fig. 5 (a). As can be seen, a small cut value implies a small gap value, but a small gap value can go together with any kind of a cut value. Fig. 6 contains the same points as 5 (a), but the marker sizes reflect the number of vertices in the graphs that are being split. Splits in big graphs tend to result in the smallest cut values.

In Fig. 5 (b), we have the same cut-gap pairs as in Fig. 5 (a). Each cut-gap pair plotted as a blue star corresponds to a good split of some graph — this time, we have also computed the cut and the gap values for a random split of the

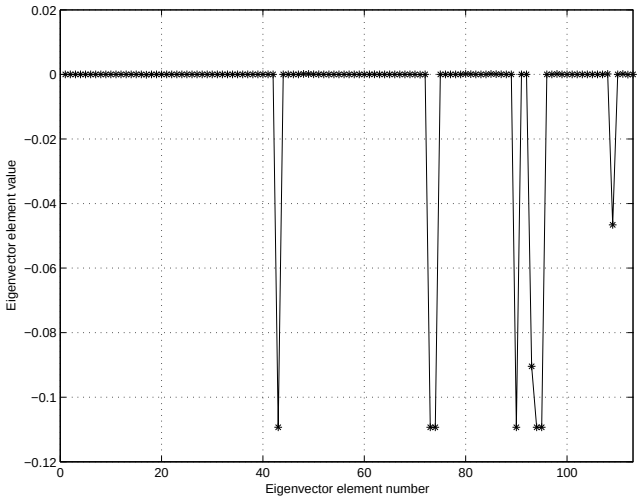


Figure 3: The 2nd eigenvector of P for the `talk` hierarchy.

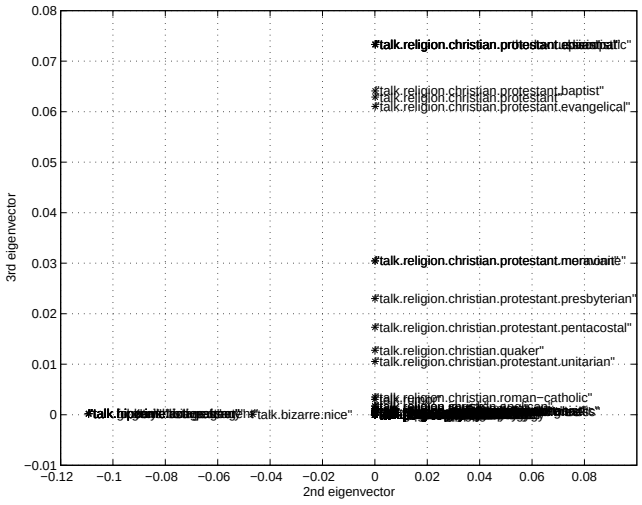


Figure 4: The 2nd and the 3rd eigenvector of P for the `talk` hierarchy.

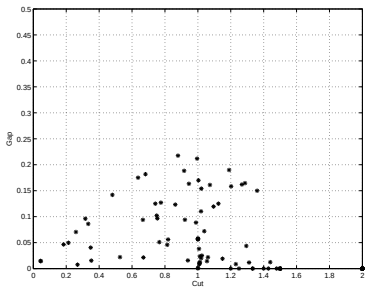
same graph, and plotted the resulting cut-gap pair as a red circle. To produce the random split, we have chosen the sizes of the two clusters at random, and assigned the data points into the two clusters at random.

Figs. 5 (d), (e), and (f) show some evidence on that the cut value is indeed a good choice for pruning the dendrogram. For instance, one might initially think $\min \text{cut}/ < \text{cut} >$ would be a better choice, since the cut values have some dependence on the size of the graph being split, and scaling might be necessary. In this expression, $< \text{cut} >$ is the average cut value over all possible splits of the graph. However, Fig. 5 (d) shows that pruning based on $\min \text{cut}$ and $\min \text{cut}/ < \text{cut} >$ would lead to a very similar result.

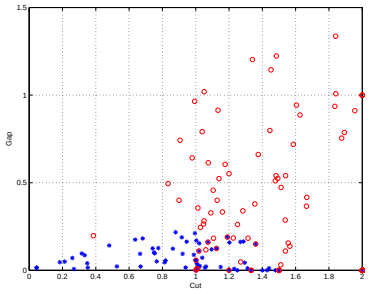
Since the hierarchical clustering dendrogram is too big to visualize, we display only the top five split levels in Fig. 7. For each split, we have displayed the cut, the gap, and the eigengap, together with the sizes of the resulting clusters. The leftmost cluster is always the larger one. For instance, we start by splitting the set of 113 newsgroups into clusters of sizes 105 and 8; this split is associated with cut value of 0.04, gap value of 0.01, and eigengap value of 0.00. The small eigengap implies that the split is highly unstable — in this case, there are more than two clear clusters in the data, and there are several possible places for the first split. The small gap implies that the second eigenvector is almost piecewise constant, which can be confirmed by looking at Fig. 3. Finally, the small cut implies that the two clusters are well separated in the normalized cut sense.

The leaves of the dendrogram are annotated with cluster number, cluster size, and an example of a newsgroup in the cluster. Let us now look at the dendrogram in more detail. On the right-hand side on level 4, we have a split our data into a cluster of size 6 and another cluster of size 2 with reasonably high cut and gap values (0.92 and 0.19, respectively). Interestingly, all consecutive splits have a cut value of 0. The cluster of size 2 (containing the result clusters 4 and 8) consists of two connected points, and splitting this kind of a cluster into two individual points results in a maximum value for a cut (2) and a minimum value for the gap (0). The eigengap cannot be determined because we only have two eigenvalues, but for convenience, we have plotted '0' for the eigengap value in this case. The newsgroups in the cluster of size 6 (containing the result clusters 2, 10, 6, and 13) happen to form a complete graph. Splitting a complete graph results in high cut values and zero gap values, as can be seen in the dendrogram.

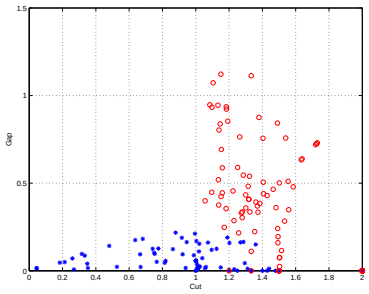
Fig. 8 shows the dendrogram for the talk hierarchy computed with the constraint that we only perform a split if the cut value is less than 0.4. We have displayed the entire dendrogram, since it is not very wide or deep this time. The contents of the smallest leaf clusters are listed in Table 7.



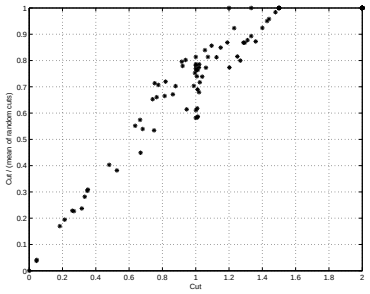
(a) All 112 min cut - gap pairs.



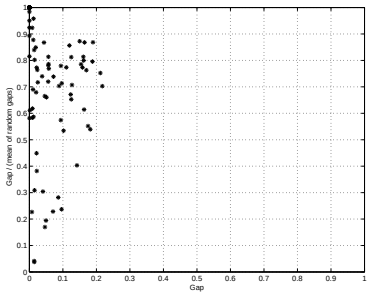
(b) All 112 min cut - gap pairs (blue stars), each of them with a random cut-gap pair (red circles) for the same data points.



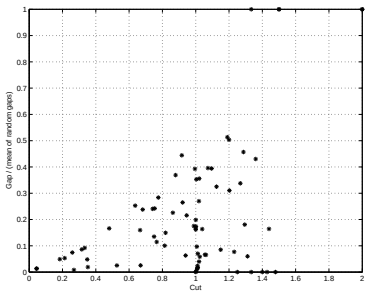
(c) All 112 min cut - gap pairs (blue stars), each of them with a mean (red circles) of 50 random cut-gap pairs for the same data points.



(d) Min cut vs. (min cut) / (mean of 50 random cuts for the same data points).



(e) (Gap corresponding to min cut) vs. (gap corresponding to min cut) / (mean of 50 random gaps for the same data points).



(f) (Min cut) vs. (Gap corresponding to min cut) / (mean of 50 random gaps for the same data points).

Figure 5: Cuts and gaps in the `talk` data.

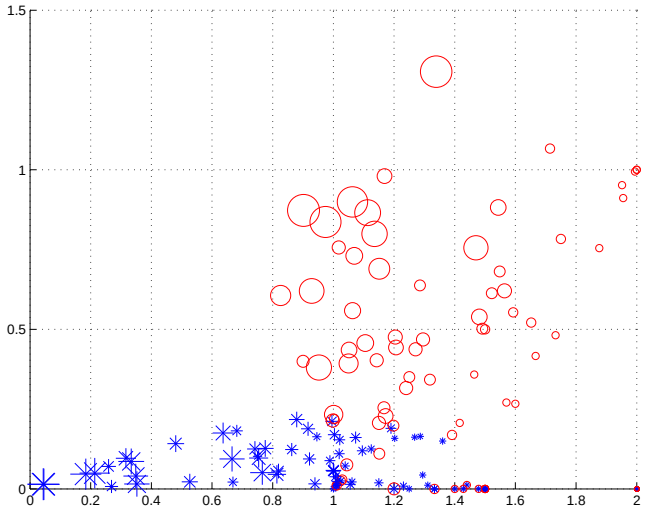


Figure 6: All 112 min cut - gap pairs (blue stars) for `talk` data, each of them with a random cut-gap pair (red circles) for the same data points. The size of the marker represents the number of vertices in the graph being cut. These sizes vary between 2 and 113.

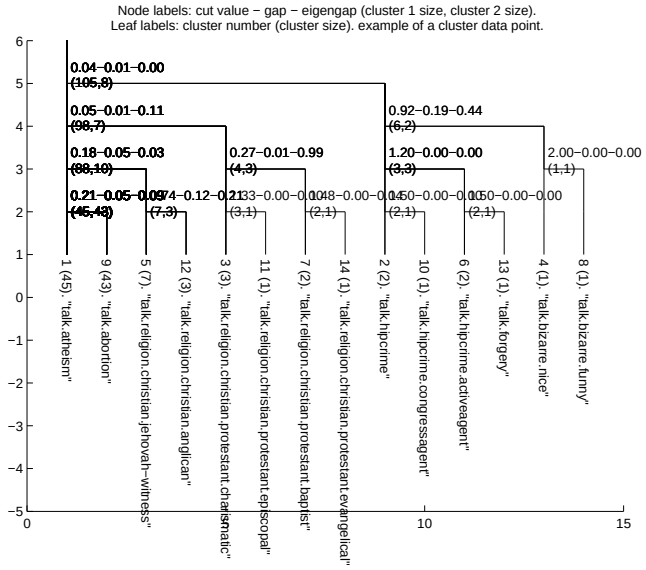


Figure 7: Five first split levels of the newsgroup dendrogram for the `talk` hierarchy.

Cluster ID	Cluster contents
2	talk.hipcrime.activeagent, talk.forgery, talk.gibberish.bill-palmer, talk.hipcrime, talk.bizarre.funny, talk.hipcrime.congressagent, talk.hipcrime.listagent, talk.bizarre.nice.
7	talk.religion.christian.protestant.baptist, talk.religion.christian.protestant.evangelical, talk.religion.christian.protestant.
3	talk.religion.christian.protestant.charismatic, talk.religion.christian.orthodox.russian, talk.religion.christian.protestant.episcopal, talk.religion.christian.protestant.adventist.
5	talk.religion.christian.jehovah-witness, talk.religion.christian.anglican, talk.religion.christian.apostolic, talk.religion.christian.arian, talk.religion.christian.coptic, talk.religion.christian.orthodox.greek, talk.religion.christian.mormon, talk.religion.christian.orthodox, talk.religion.christian.orthodox.misc, talk.religion.christian.nestorian.
21	talk.politics.china, talk.politics.tibet.
15	talk.religion.christian.protestant.pentacostal, talk.religion.christian.protestant.presbyterian, talk.religion.christian.quaker, talk.religion.christian.protestant.mennonite, talk.religion.christian.protestant.moravian, talk.religion.christian.protestant.unitarian.
23	talk.religion.christian.science, talk.religion.confucianism, talk.religion.jewish.conservative, talk.religion.jewish.messianic, talk.religion.satanism, talk.religion.scientology, talk.rumor, talk.religion.shinto, talk.underwear.veg, talk.religion.jewish.reconstructionist, talk.religion.jewish.reform, talk.religion.rosicrucian, talk.religion.tao, talk.religion.zoroastrian, talk.religion.jewish.orthodox, talk.religion.jewish.orthodox.chassidic, talk.religion.jewish.

Table 7: Selected clusters in the pruned `talk` dendrogram (from right to left).

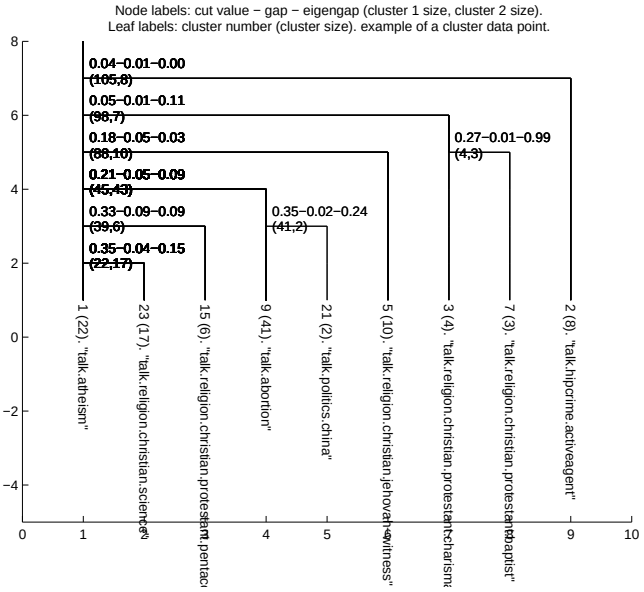


Figure 8: Dendrogram for the `talk` data with $cut < 0.4$. There are 11 splits that satisfy this condition; all of them are included in the dendrogram.

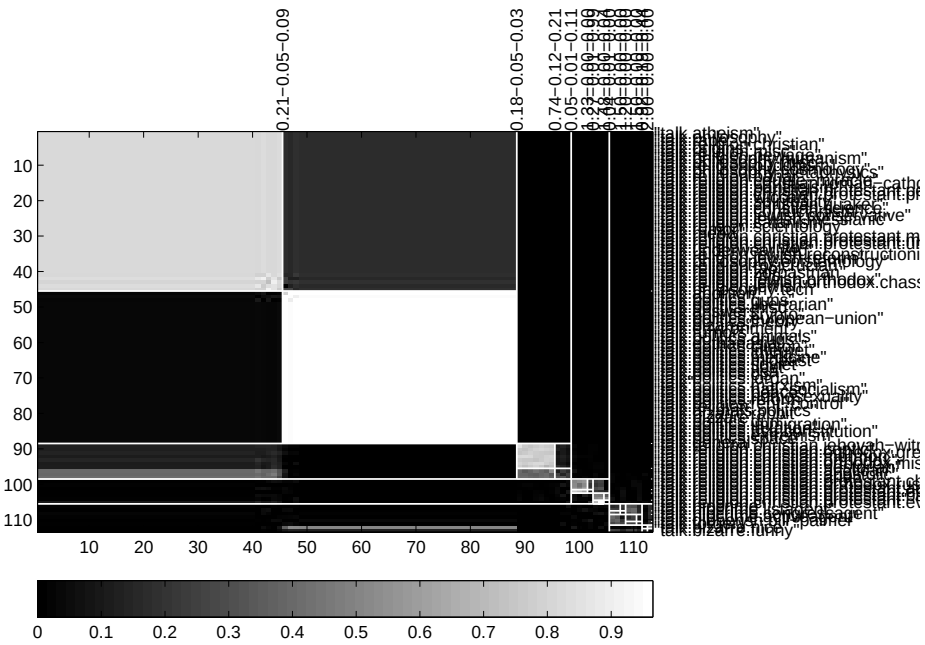


Figure 9: Block dendromatrix for the hierarchical clustering of the `talk` newsgroups.

7.2.1 Dendromatrices

Let us now try out an alternative way for visualizing our spectral hierarchical clustering results. We have constructed dendromatrices showing the 5 topmost split levels of the original hierarchical clustering (for the similarity matrix with zero diagonal) for the 113 `talk` newsgroups; see Fig. 9 and Fig. 10. The size of a dendromatrix is (number of newsgroups N) $\times$ (number of newsgroups N), in this case 113×113. The red horizontal and vertical lines represent the division of the data points into clusters. The values on the x- and y-axis are just the newsgroup indices. The newsgroup names are visible on the right. The top part of the image shows the cut, the gap, and the eigengap values for each split.

We have plotted the block dendromatrix for the `talk` data in Fig. 9. Many of the diagonal blocks in the block dendromatrix are light gray, indicating that a random walk starting in a cluster is likely to remain there. On the other hand, most non-diagonal blocks are dark in color, showing the low probability of moving from a cluster to another.

Alternatively, we can form a *unit dendromatrix*, an example of which is shown in Fig. 10. The unit dendromatrix reveals that, while most of the light gray dots are located in the diagonal blocks, a large part of the diagonal block entries are black. This implies that there is an intricate and complicated inner

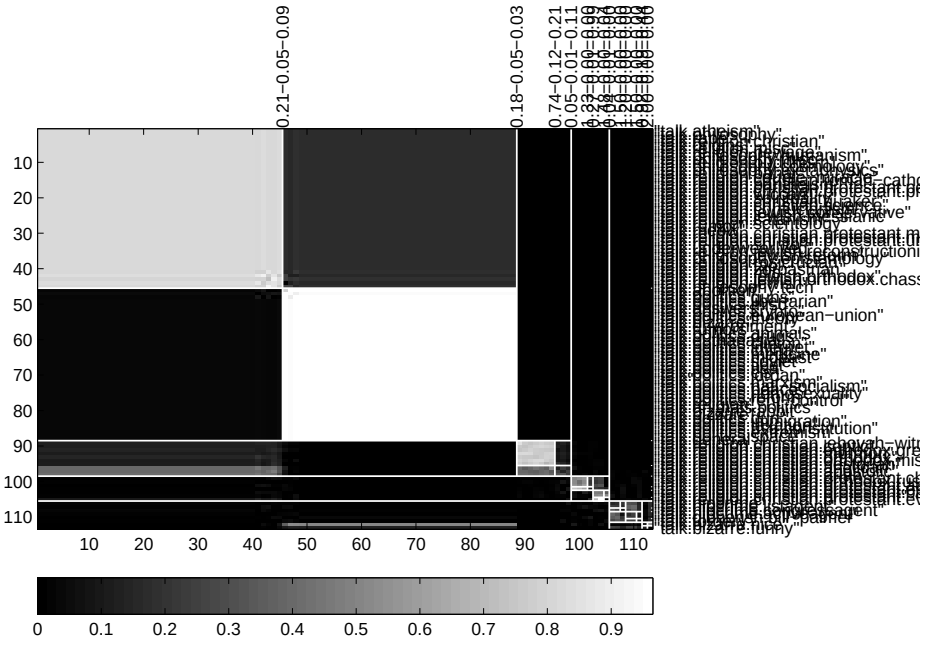


Figure 10: Unit dendromatrix for the hierarchical clustering of the `talk` newsgroups.

structure within each cluster.

7.2.2 Non-Zero Diagonal in the Similarity Matrix

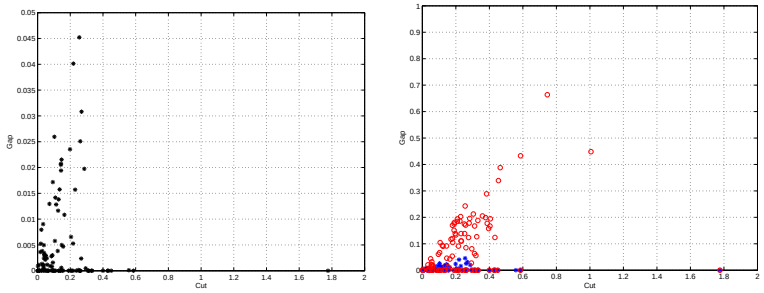
In the previous experiments, the diagonal of the similarity matrix has been set to zero. Let us now investigate what happens if we insert the total number of postings in each newsgroup in the diagonal. Note that we have the total number of postings in the period between January 2000 and December 2003, but the crossposting data is only between January 2000 and May 2003. We do not have the total number of postings available for the following groups:

`talk.religion.christian.protestant.baptist`,
`talk.religion.christian.protestant.charismatic`,
`talk.religion.christian.protestant.pentacostal`,
`talk.religion.christian.protestant.presbyterian`,
`talk.religion.christian.protestant.evangelical`,
`talk.religion.christian.protestant.mennonite`,
`talk.religion.christian.orthodox.russian`,
`talk.religion.christian.protestant.moravian`,
`talk.religion.christian.protestant.unitarian`,
`talk.religion.christian.protestant.episcopal`,
`talk.religion.christian.protestant.adventist`,
`talk.general`.

We remove these 12 groups from the matrix; 101 newsgroups remain. It is worth noting that the difference between the total number of postings and the total number of crosspostings is negative for 19 newsgroups; that is, in some cases there are more crosspostings than original postings. The range of the differences falls between -3500 and 971,899.

Note also that our similarity matrix is formed of start and reply crosspostings only; we have left barrens out of considerations. However, the total number of postings we have inserted on the diagonal of the similarity matrix contains starts, replies, and barrens. Therefore the clustering results on this similarity matrix should be taken only as a preliminary example of what kind of results are expected when the diagonal contains non-zero values.

We re-run spectral hierarchical clustering for the 101 newsgroups. Due to the large values on the diagonal, the cut values have decreased significantly. If we prune the dendrogram with the cut threshold of 0.4 as before, we end up with 95 splits instead of 8 splits. See Fig. 11 for the distribution of the cut and the gap values. We decide to prune dendrogram with the cut threshold of 0.03. The resulting dendrogram is shown in Fig. 12 and a listing of the contents of the selected leaf clusters can be found in Table 8.



(a) All 100 (min cut)-gap pairs.

(b) All 100 (min cut)-gap pairs (blue stars) with random cut-gap pairs (red circles).

Figure 11: Cut vs. gap in the `talk` data (similarity matrix with a non-zero diagonal).

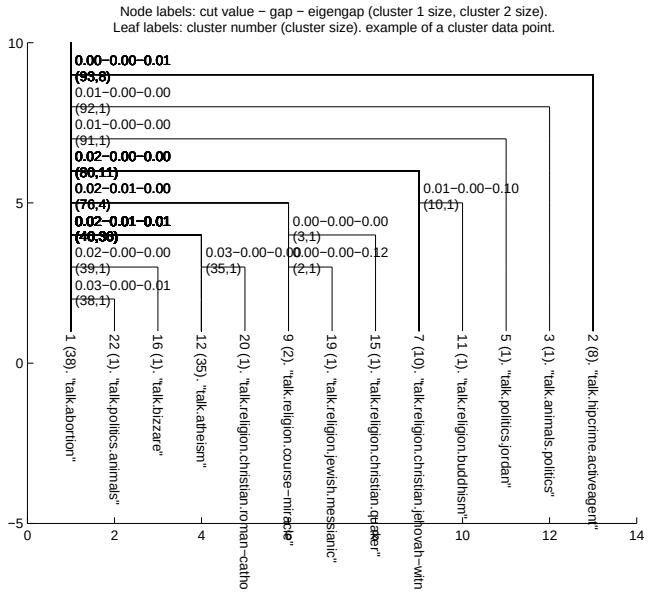


Figure 12: Dendrogram for the `talk` data with $cut < 0.03$, formed using a similarity matrix with a non-zero diagonal.

Cluster ID	Cluster contents
2	"talk.hipcrime.activeagent", "talk.forgery", "talk.gibberish.bill-palmer", "talk.hipcrime", "talk.bizarre.funny", "talk.hipcrime.congressagent", "talk.hipcrime.listagent", "talk.bizarre.nice".
3	"talk.animals.politics".
5	"talk.politics.jordan".
11	"talk.religion.buddhism".
7	"talk.religion.christian.jehovah-witness", "talk.religion.christian.anglican", "talk.religion.christian.apostolic", "talk.religion.christian.arian", "talk.religion.christian.coptic", "talk.religion.christian.orthodox.greek", "talk.religion.christian.mormon", "talk.religion.christian.orthodox", "talk.religion.christian.orthodox.misc", "talk.religion.christian.nestorian".
15	"talk.religion.christian.quaker".
19	"talk.religion.jewish.messianic".
9	"talk.religion.course-miracle", "talk.religion.wiccan".
20	"talk.religion.christian.roman-catholic".
16	"talk.bizzare".
22	"talk.politics.animals".

Table 8: Selected clusters in the `talk` dendrogram, formed with a similarity matrix with a non-zero diagonal (from right to left).

7.3 Stability of Talk Dendrograms

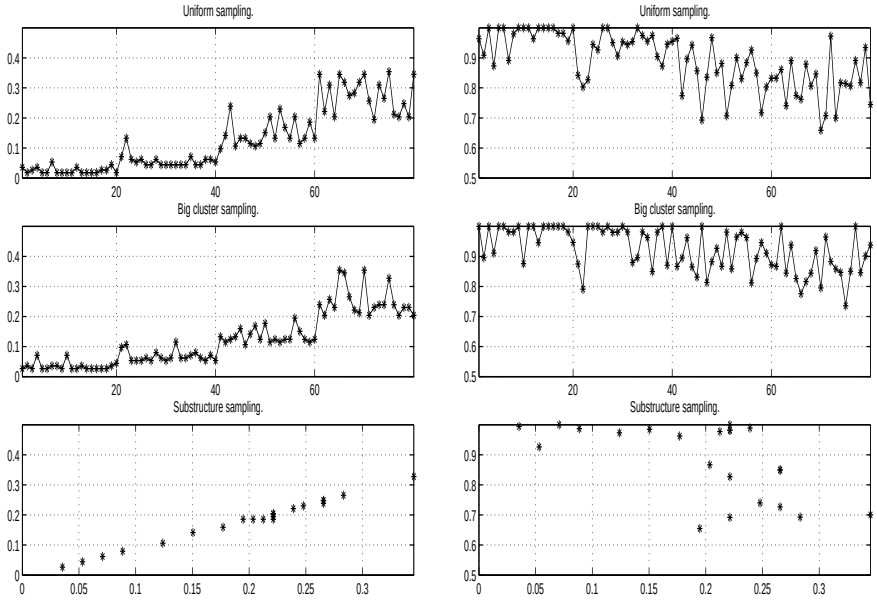
We will evaluate the stability of the dendrogram in Fig. 8 by sampling the set of 113 `talk` newsgroups. The choose to use sample percentages 0.025, 0.05, 0.10, and 0.20 (the portion of data points we remove). We sample the data with each percentage 20 times. We run spectral hierarchical clustering on each sample and obtain 4×20 new dendrograms. The original dendrogram has been formed by pruning with cut threshold 0.4. We use the same cut threshold for pruning the dendrograms on the samples. We then compare each sample dendrogram with the original dendrogram to find out how stable the original dendrogram is. We repeat this process for each of the three types of sampling introduced in Section 5.1: uniform sampling, big cluster sampling, and substructure sampling.

As described in Section 5.2, there are several different ways to compare hierarchical clusterings. In absence of sufficient research on the various methods, we try out several alternatives. We will compare leaf clusterings with Clustering Error (CE) [8]. We will compare the dendrograms with Subspace Clustering Error (SCE) [11]. We will form both unweighted and weighted cophenetic matrices. The weighted cophenetic matrices are computed using dendrogram edge weights $exp(-1.5 * \text{cut})$. We will compare both types of matrices with correlation coefficient, L1 distance between matrix elements, and L2 distance between matrix elements. The correlation coefficient does not depend on the scaling/normalization of the cophenetic matrices. However, the scaling/normalization affects the L1 and the L2 distances. We will run experiments with no scaling, scaling with N, scaling with $log_2(N)$, L1 normalization, and L2 normalization (see Section 5.2 for details; additional results are presented in [1]).

Taking into account the 3 sampling methods, 5 scaling/normalization options, 5 ways to compare clusterings, and the possibility of weighting the cophenetic matrix, we have a total of 72 different ways to evaluate the stability of a hierarchical clustering. We cannot present all these results here. Instead, we present the most interesting subset of the results and comment on the rest of the results briefly.

Fig. 13 (a) presents the Clustering Error results. Uniform sampling and big cluster sampling behave in a very similar way — error and the variance of the error increase as more data points are removed. The CE values in the two graphs are strikingly close to each other. Note that in the substructure sampling graph, we gave removed up to 35% of the data points, as the substructure sampling amounts are rarely exactly those desired. Also in the substructure sampling graph, the clustering error increases with the percentage of points removed.

See Fig. 13 (b) for the correlation between weighted cophenetic matrices. The results for correlation between unweighted cophenetic matrices are almost the same, so we have chosen not to plot those. The correlation does not depend on the scaling or the normalization of the cophenetic matrices. In uniform sampling and big cluster sampling, the correlation shows a decreasing trend, but much weaker than in the case of CE. The substructure sampling graph reveals that removing about 19% of data points or more in form of substructures has potentially a significant effect on the structure of the dendrogram.



(a) Clustering Error.

(b) Correlation between weighted cophenetic matrices.

Figure 13: Stability of the dendrograms for `talk` data. The sample percentages are 0.025, 0.05, 0.10, and 0.20. We have sampled 20 times with each percentage. That is, the points 1–20 represent the samples with sample percentage 0.025, the points 21–40 represent the samples with sample percentage 0.05, and so on. In each subfigure, the topmost plot is for uniform sampling, the middle plot is for big cluster sampling, and the bottom plot is for substructure sampling. In the uniform and big cluster sampling plots, we have the sample number on the x-axis. In the substructure sampling plot, the x-axis represents the sample percentage.

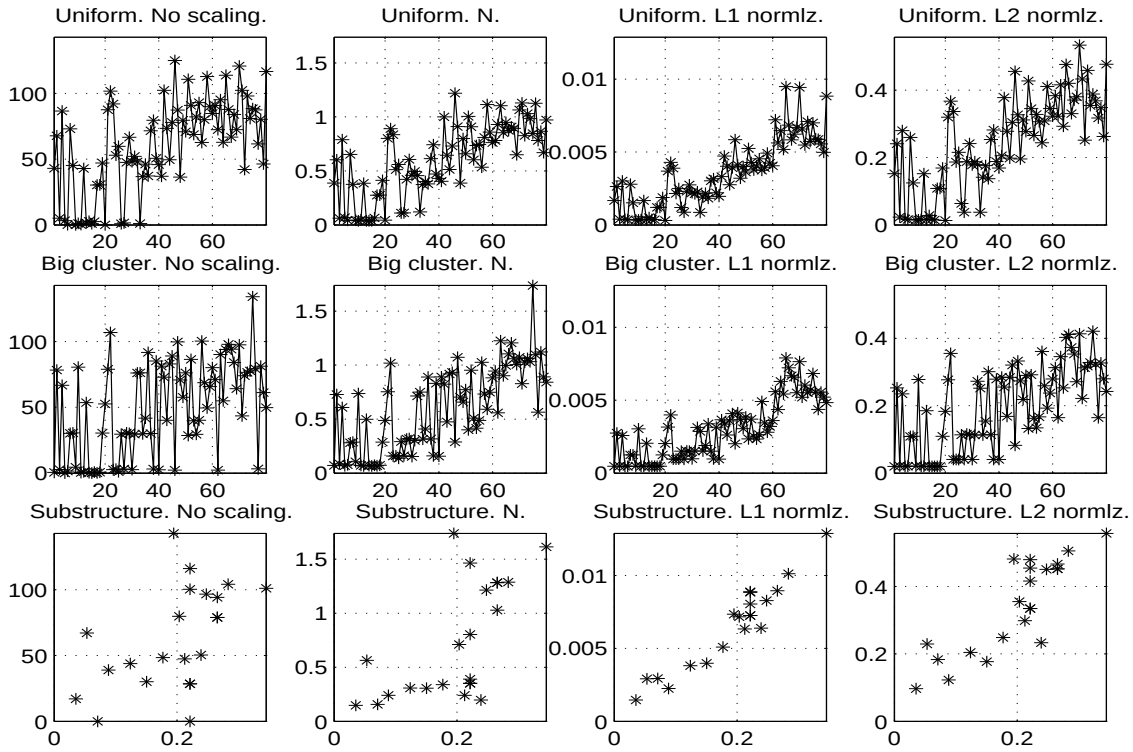


Figure 14: Stability of the dendrograms for `talk` data. L2 distance between weighted cophenetic matrices. The sample percentages are 0.025, 0.05, 0.10, and 0.20. We have sampled 20 times with each percentage. That is, the points 1–20 represent the samples with sample percentage 0.025, the points 21–40 represent the samples with sample percentage 0.05, and so on. Top row: Uniform sampling. Middle row: Big cluster sampling. Bottom row: Substructure sampling. Left column: No scaling of cophenetic matrices. Second column: Cophenetic matrices scaled by N. Third column: Cophenetic matrices with L1 normalization. Right column: Cophenetic matrices with L2 normalization. In the uniform and big cluster sampling plots, we have the sample number on the x-axis. In the substructure sampling plots, the x-axis represents the sample percentage.

Let us now consider the L1 and the L2 distances between the cophenetic matrices. All results for unweighted and weighted cophenetic matrices are very similar, so we choose to plot only those for weighted matrices. The L1 and the L2 results do not differ much either. Since the L2 results display slightly clearer trends with respect to the sample sizes, we choose to plot only L2 results. Let us now turn to consider the scaling/normalization of the cophenetic matrices. In Fig. 13, we have plotted the results for unscaled matrices as a baseline. We have observed that scaling by $log(N)$ does not have much effect, so we will not display those results. Instead, scaling by N and L1 and L2 type normalizations result in some beautiful trends in the graphs, so we will display those results in Fig. 13.

When we look at Fig. 13 we notice that the L1 normalization results in the clearest upward trend with the sample percentage in all three sampling schemes. Also L1 normalization and scaling by N result in an upward trend, but the variance of the error is larger. Judging by the strength of the upward trend and the size of the variance, not scaling the cohpenetic matrices produces worst results.

Comparing all of the above results, we can conclude that sampling seems to have larger effect on the tree structure (cophenetic matrix based comparisons) than on the leaf clusters (CE based comparisons). In other words, if we sample several times using a fixed sample percentage p, we are likely to end up with reasonably similar leaf clusters, but these leaf clusters might be located in varying places in the tree. If we wish to compare leaf clusterings, CE distance with any type of sampling seems like a good choice. But if we wish to compare the tree structures, the most reliable methods seem to be correlation between weighted cophenetic matrices and L2 distance between weighted cophenetic matrices with L1 normalization, using either uniform or big cluster sampling.

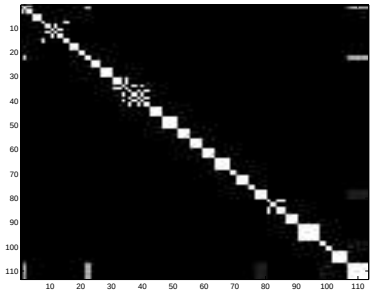
7.4 Natural Communities in Talk Data

Let us explore the natural communities in the `talk` by means of the mean connectivity matrix, described in Section 6. To construct the mean connectivity matrix, we need to decide upon the number of clusters K and the number of K-means runs. We would hope that running the K-means algorithm thousands of times for a wide range of values of K would result in a 'true picture' of the data. Unfortunately, our experiments have demonstrated that the structure of the mean connectivity matrix depends greatly on the parameters we have chosen.

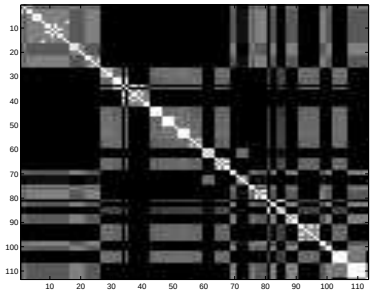
To illustrate this, we have plotted four mean connectivity matrices in Fig. 7.4. Fig. 7.4(c) is the result of 100,000 K-means runs for $K \in \{10, 15, 20, 25, 30, 35, 40\}$. The maximum number of K-means iterations per run was set to 500. Re-running the experiment with these parameter values results in almost exactly same mean connectivity matrix; in general, we have observed that thousands or even tens of thousands of K-means runs are required for a stable result. Even then, varying the values of K has a some effect on the results, as Fig. 7.4 (d) demonstrates. If we have only 1000 K-means runs, the effect of K values is very large, as can be seen in Figs. 7.4 (a) and 7.4 (b).

Let us now look at the mean connectivity matrix of Fig. 7.4 (c) in more detail. The matrix clearly contains several stable clusters (white blocks on the diagonal). The contents of seven of these white blocks in the middle of the matrix are listed in Table 7.4. Interestingly, only some of these groups form clusters in our spectral hierarchical dendrogram (see Fig. 8 and Table 7). In four of the seven groups below only two or three newsgroups are clustered together in the dendrogram.

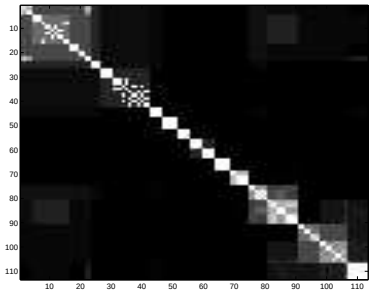
An interesting avenue to explore in the future is to see if the mean connectivity matrices are able to reveal information on the hierarchical clustering of the newsgroups; including both small and large values of K in the experiments could result in hierarchical structure in the mean connectivity matrix. These results could then be compared to the dendrograms produced by divisive hierarchical clusterings.



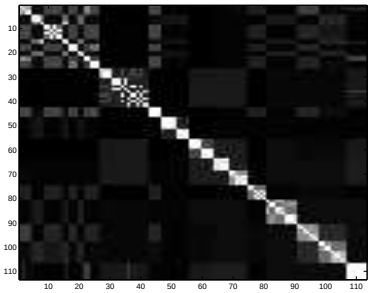
(a) 1000 K-means runs for $K = 30$.



(b) 1000 K-means runs for $K \in \{10, 50\}$.



(c) 100,000 K-means runs for $K \in \{10, 15, 20, 25, 30, 35, 40\}$



(d) 100,000 K-means runs for $K \in \{6, 13, 19, 26, 33, 48, 55\}$

Figure 15: Mean connectivity matrices for the `talk` newsgroups. The matrices show the average result of several K-means clusterings. The entry (i, j) represents the fraction of the clusterings in which the ith and the jth newsgroup are in the same cluster. White depicts '1' and black depicts '0'. The ordering of the rows and the columns is the same in all four matrices.

talk.politics.usa, talk.abortion, talk.environment, talk.euthanasia
talk.philosophy, talk.religion.christian, talk.philosophy.misc, talk.religion, talk.meow
talk.politics.marxism, talk.politics.natl-socialism, talk.politics.reform, talk.politics.rent-control
talk.religion.christian.nestorian, talk.religion.christian.jehovah-witness, talk.religion.christian.coptic, talk.religion.christian.orthodox.greek
talk.politics.clinton, talk.politics.internet, talk.politics.peace, talk.politics.extremism
talk.religion.christian.science, talk.religion.confucianism, talk.religion.jewish.conservative, talk.religion.jewish.messianic, talk.religion.jewish.orthodox
talk.hipcrime.activeagent, talk.hipcrime.congressagent, talk.forgery, talk.gibberish.bill-palmer, talk.hipcrime, talk.hipcrime.listagent

Table 9: Some natural communities in the `talk` data. Newsgroup clusters corresponding to the seven middle blocks of the mean connectivity matrix in Fig. 7.4 (a).

8 Spectral Hierarchical Clustering for Other Data Sets

Table 8 contains a summary of the data sets for which we have computed hierarchical clusterings. The similarity matrices are formed of starts and replies (no barrens) with zeros on the diagonal. For each data set, we have clustered only the biggest connected component. We have computed the spectral hierarchical clustering and decided upon the pruning threshold based on the cut-gap graph. We have displayed the cut-gap graph and the dendrogram, and listed the contents of the small and medium-sized leaf clusters. We have evaluated the stability of the dendrograms in several different ways with the exception of the three biggest data sets. We have not presented all of our stability results here; for the rest of the results, refer to [1].

Hierarchies	**Sizes of connected components**	**Cut-gap graph**	**Dendrogram**	**Clusters**	**Stability**
`us`	95, 2[4]	Fig. 16	Fig. 17	Table 11	Fig. 18, Fig. 19
`talk`	113	Fig. 5	Fig. 8	Table 7	Fig. 13, Fig. 14
`alt.religion`, `soc.religion`, `talk.religion`	320	Fig. 20	Fig. 21	Table 12	Fig. 22, Fig. 23
`uk`	522	Fig. 24	Fig. 25	Table 13	Fig. 26, Fig. 27
`aus, es`	536	Fig. 28	Fig. 29	Table 14	Fig. 30, Fig. 31
`soc, talk`	540, 2[5]	Fig. 32	Fig. 33	Table 15	Fig. 34, Fig. 35
`uk, us`	620	Fig. 36	Fig. 37	Table 16	Fig. 38, Fig. 39
`rec`	1138	Fig. 40	Fig. 41	Table 17	–
`soc, comp, sfnet`	2114, 2, 2, 2[6]	Fig. 42	Fig. 43	Table 18	–
`microsoft`	3010, 5[7]	Fig. 44	Fig. 45	Table 19	–

Table 10: Data sets with hierarchical clusterings.

[4] `us.sport.football.pro, us.sport.football.misc`

[5] `soc.frogbutt, talk.frogbutt`

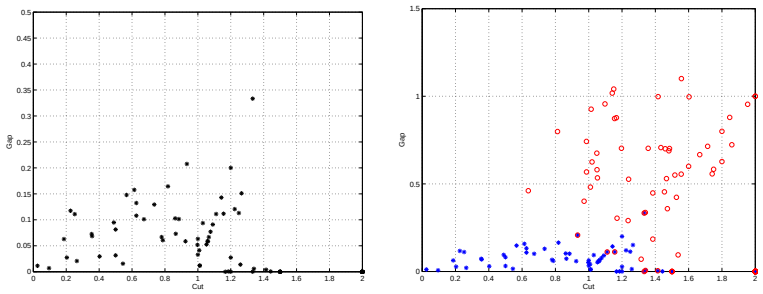
[6] `comp.patents, soc.history.property` and `sfnet.lists.lpf, sfnet.lists.nysersnmp` and `comp.frogbutt, soc.frogbutt`

[7] `microsoft.public.br.asp, microsoft.public.br.dotnet, microsoft.public.br.dotnet.framework, microsoft.public.br.dotnet.framework.adonet,`

All cut-gap graphs reveal an interesting phenomenon: some of the poins seem to lie on a line. In fact, there are two lines, 'gap = cut-1' and 'gap = 2(cut-1.5)'. A close inspection of the points on the lines reveals that the points on the line 'gap = 2(cut-1.5)' correspond to random splits of graphs with three nodes. At (cut = 1.5, gap = 0), we have a complete graph of size three (triangle) with equal edge weights. At (cut = 2, gap = 1), one of the edges of the triangle has weight 0. And in between those points along the line, we have triangles of various edge weights.

The points on the line 'gap = cut-1' correspond to random splits of complete graphs with 3, 4, or 5 nodes and various edge weights. There is sometimes also a second line 'gap = cut-1' formed of minimum cut splits, instead of random splits. Those seem to be splits on star-shaped graphs.

`microsoft.public.br.dotnet.framework.aspnet`



(a) All 94 (min cut)-gap pairs.

(b) All 94 (min cut)-gap pairs (blue stars) with random cut-gap pairs (red circles).

Figure 16: Cut vs. gap in the `us` data.

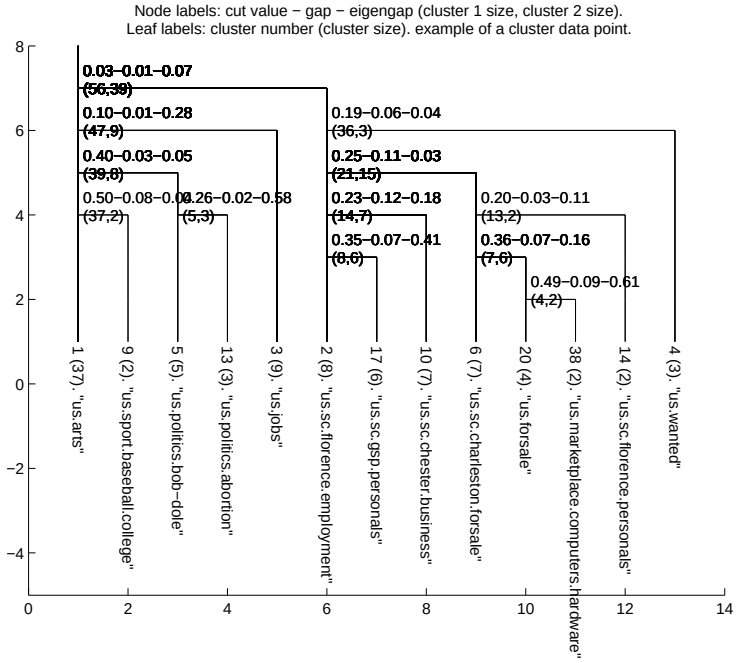
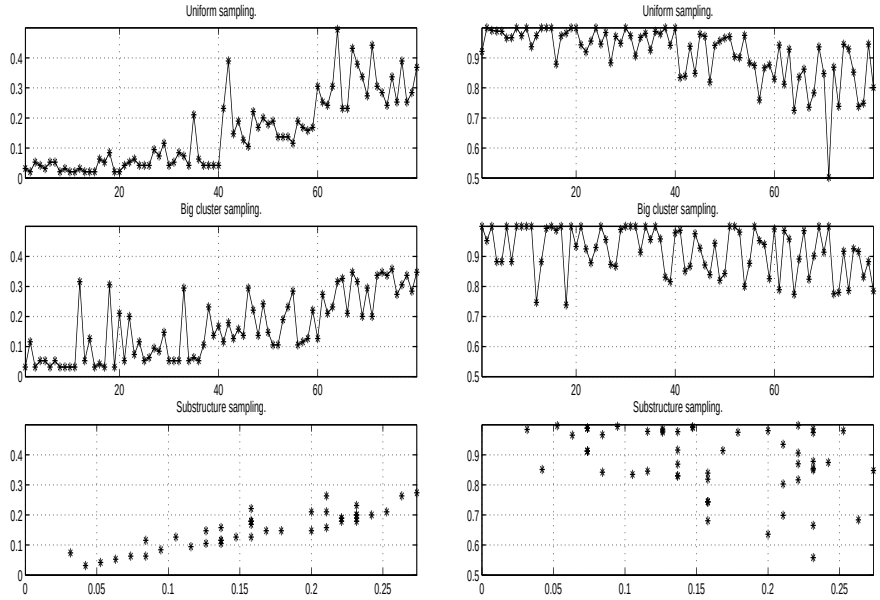


Figure 17: Dendrogram for the `us` data with $cut < 0.5$. The low cut threshold have been chosen for the visualization purposes.

Cluster ID	Cluster contents
4	us.wanted, us.wanted.misc, us.wanted.d.
14	us.sc.florence.personals, us.sc.lancaster.personals.
38	us.marketplace.computers.hardware, us.marketplace.computers.software.
20	us.forsale, us.forsale.computers, us.forsale.d, us.forsale.misc.
6	us.sc.charleston.forsale, us.sc.chester.forsale, us.sc.columbia.forsale, us.sc.florence.forsale, us.sc.gsp.forsale, us.sc.lancaster.forsale, us.sc.rockhill.forsale.
10	us.sc.chester.business, us.sc.charleston.personals, us.sc.rockhill.personals, us.sc.charleston.dining, us.sc.charleston.talk, us.sc.chester.talk, us.sc.columbia.talk.
17	us.sc.gsp.personals, us.sport.football.college, us.sc.gsp.politics, us.sc.gsp.wanted, us.sc.gsp.talk, us.sc.lancaster.dining.
2	us.sc.florence.employment, us.sc.lancaster.employment, us.sc.rockhill.employment, us.sc.charleston.business, us.sc.florence.business, us.sc.gsp.business, us.sc.lancaster.business, us.sc.rockhill.business.
3	us.jobs, us.jobs.contract, us.jobs.misc, us.jobs.offered, us.jobs.offered.contract, us.jobs.resumes, us.sc.columbia.employment, us.jobs.resume, us.resumes.jobs.
13	us.politics.abortion, us.sc.charleston.politics, us.issues.abortion.
5	us.politics.bob-dole, us.msis.general, us.politics.phil-gramm, us.rec.scouting, us.sc.charleston.
9	us.sport.baseball.college, us.sport.baseball.

Table 11: Selected clusters in the us dendrogram (from right to left).



(a) Clustering Error.

(b) Correlation between weighted cophenetic matrices.

Figure 18: Stability of the dendrograms for `us` data. The sample percentages are 0.025, 0.05, 0.10, and 0.20. We have sampled 20 times with each percentage. That is, the points 1–20 represent the samples with sample percentage 0.025, the points 21–40 represent the samples with sample percentage 0.05, and so on. In each subfigure, the topmost plot is for uniform sampling, the middle plot is for big cluster sampling, and the bottom plot is for substructure sampling. In the uniform and big cluster sampling plots, we have the sample number on the x-axis. In the substructure sampling plot, the x-axis represents the sample percentage.

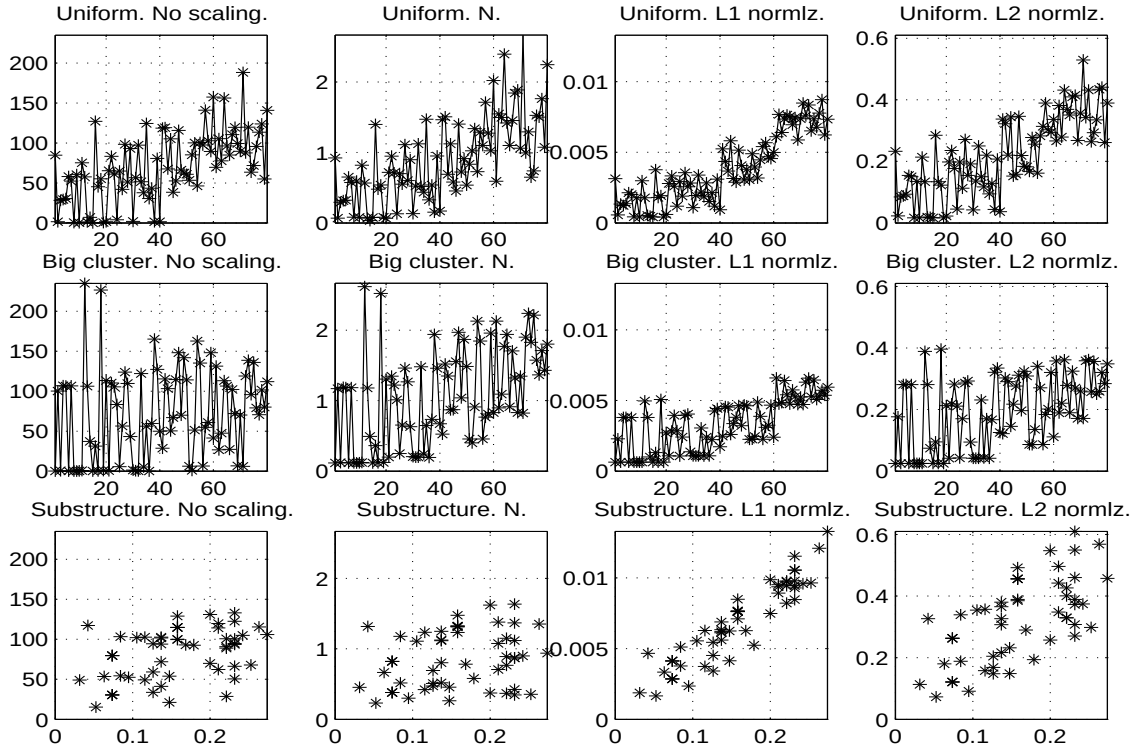
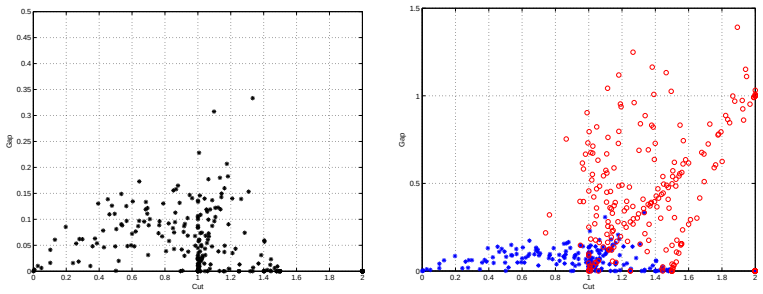


Figure 19: Stability of the dendrograms for `us` data. L2 distance between weighted cophenetic matrices. The sample percentages are 0.025, 0.05, 0.10, and 0.20. We have sampled 20 times with each percentage. That is, the points 1–20 represent the samples with sample percentage 0.025, the points 21–40 represent the samples with sample percentage 0.05, and so on. Top row: Uniform sampling. Middle row: Big cluster sampling. Bottom row: Substructure sampling. Left column: No scaling of cophenetic matrices. Second column: Cophenetic matrices scaled by N. Third column: Cophenetic matrices with L1 normalization. Right column: Cophenetic matrices with L2 normalization. In the uniform and big cluster sampling plots, we have the sample number on the x-axis. In the substructure sampling plots, the x-axis represents the sample percentage.



(a) All 319 min cut - gap pairs.

(b) All 319 min cut - gap pairs (blue stars), each of them with a random cut-gap pair (red circles) for the same data points.

Figure 20: Cuts and gaps in the `alt.religion, soc.religion, talk.religion` data.

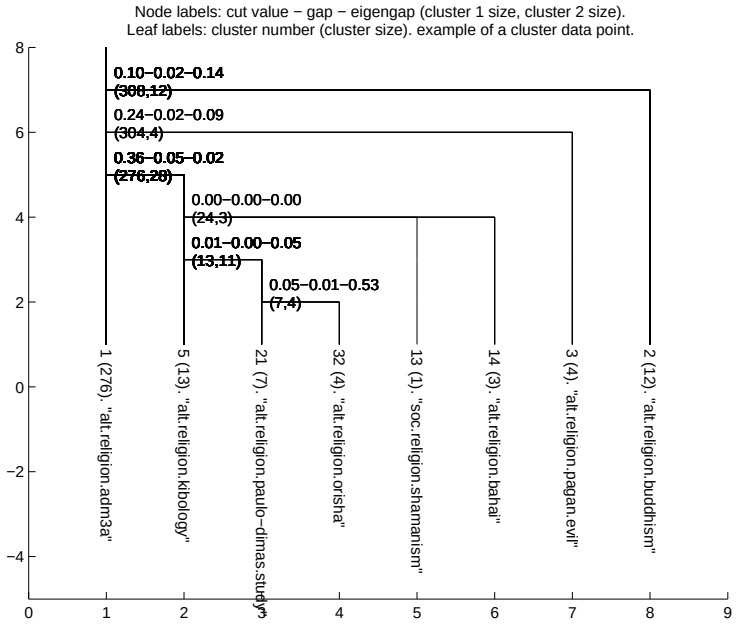
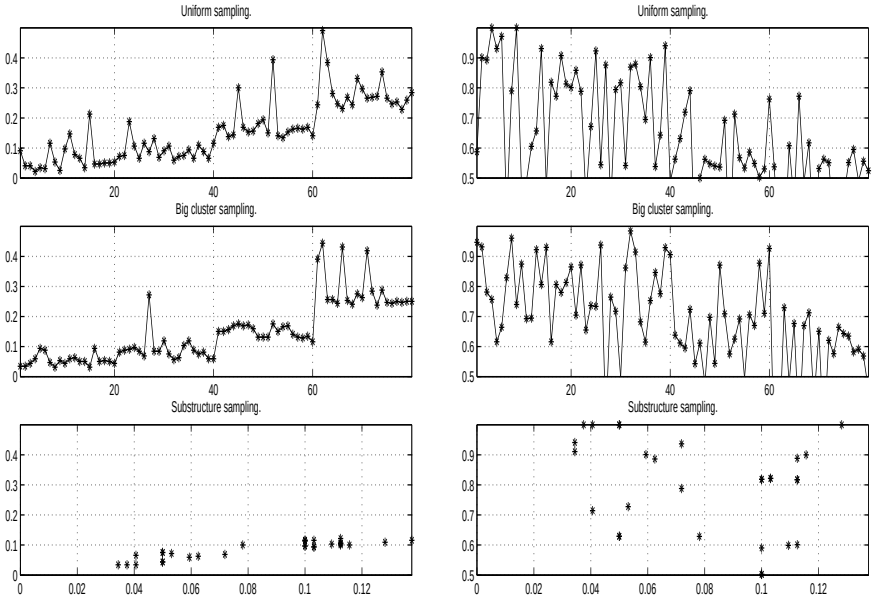


Figure 21: Dendrogram for the `alt.religion, soc.religion, talk.religion` data with $cut < 0.39$.

Cluster ID	Cluster contents
2	alt.religion.buddhism, alt.religion.buddhism.nichiren, alt.religion.buddhism.theravada, alt.religion.buddhism.tibetan, talk.religion.buddhism, alt.religion.buddhism.nkt, alt.religion.buddhism.nichiren.shoshu, alt.religion.buddhism.nichiren.shoshu.news, alt.religion.buddhism.ris-med, alt.religion.nichiren.shoshu.news, alt.religion.nichiren, alt.religion.nichiren.shoshu.
3	alt.religion.pagan.evil, alt.religion.paulo-dimas.deveras, alt.religion.pagan.nazi, alt.religion.pagan.texas.
14	alt.religion.bahai, talk.religion.bahai, soc.religion.bahai.
13	soc.religion.shamanism.
32	alt.religion.orisha, alt.religion.sabaean, alt.religion.voodoo, alt.religion.scott-mcdowell.
21	alt.religion.paulo-dimas.study, alt.religion.paulo-dimas.temples, alt.religion.paulo-dimas.worship, alt.religion.paulo-dimas.misc, alt.religion.paulo-dimas.newcomers, alt.religion.paulo-dimas, alt.religion.pcboard.
5	alt.religion.kibology, alt.religion.kibology.orthodox, alt.religion.kibology.second-coming, alt.religion.louis-nick, alt.religion.kibology.version-where-you-can-xpost-threads-about-how--much-you-hate-jesus, alt.religion.liet.santoy, alt.religion.kibology.is.dead.dead.dead, alt.religion.kibology.the-not--funny-version-where-lee-can-xpost-not-funny-stupid-threads, alt.religion.jonism, alt.religion.monica, alt.religion.kibo, alt.religion.johovahs, soc.religion.kibology.

Table 12: Selected clusters in the `alt.religion, soc.religion, talk.religion` dendrogram (from right to left).



(a) Clustering Error.

(b) Correlation between weighted cophenetic matrices.

Figure 22: Stability of the dendrograms for `religion` data. The sample percentages are 0.025, 0.05, 0.10, and 0.20. We have sampled 20 times with each percentage. That is, the points 1–20 represent the samples with sample percentage 0.025, the points 21–40 represent the samples with sample percentage 0.05, and so on. In each subfigure, the topmost plot is for uniform sampling, the middle plot is for big cluster sampling, and the bottom plot is for substructure sampling. In the uniform and big cluster sampling plots, we have the sample number on the x-axis. In the substructure sampling plot, the x-axis represents the sample percentage.

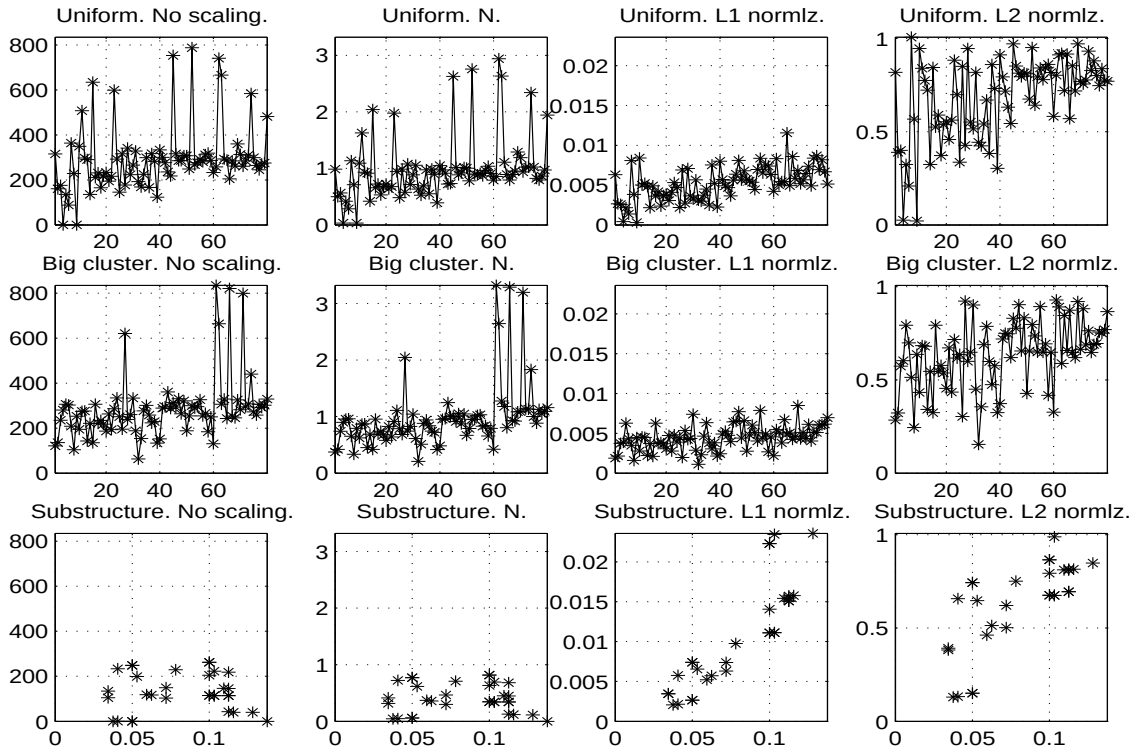
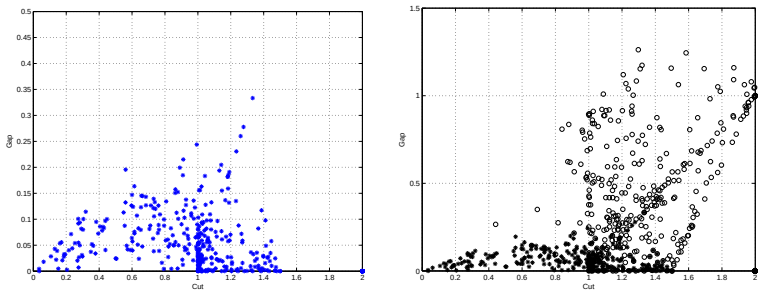


Figure 23: Stability of the dendrograms for `religion` data. L2 distance between weighted cophenetic matrices. The sample percentages are 0.025, 0.05, 0.10, and 0.20. We have sampled 20 times with each percentage. That is, the points 1–20 represent the samples with sample percentage 0.025, the points 21–40 represent the samples with sample percentage 0.05, and so on. Top row: Uniform sampling. Middle row: Big cluster sampling. Bottom row: Substructure sampling. Left column: No scaling of cophenetic matrices. Second column: Cophenetic matrices scaled by N. Third column: Cophenetic matrices with L1 normalization. Right column: Cophenetic matrices with L2 normalization. In the uniform and big cluster sampling plots, we have the sample number on the x-axis. In the substructure sampling plots, the x-axis represents the sample percentage.



(a) All 521 (min cut)-gap pairs.

(b) All 521 (min cut)-gap pairs (blue stars) with random cut-gap pairs (red circles).

Figure 24: Cut vs. gap in the `uk` data.

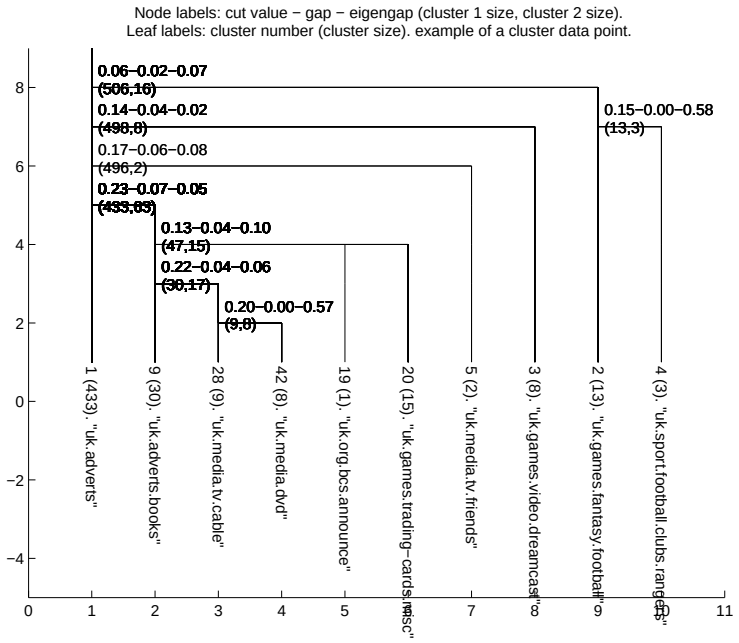
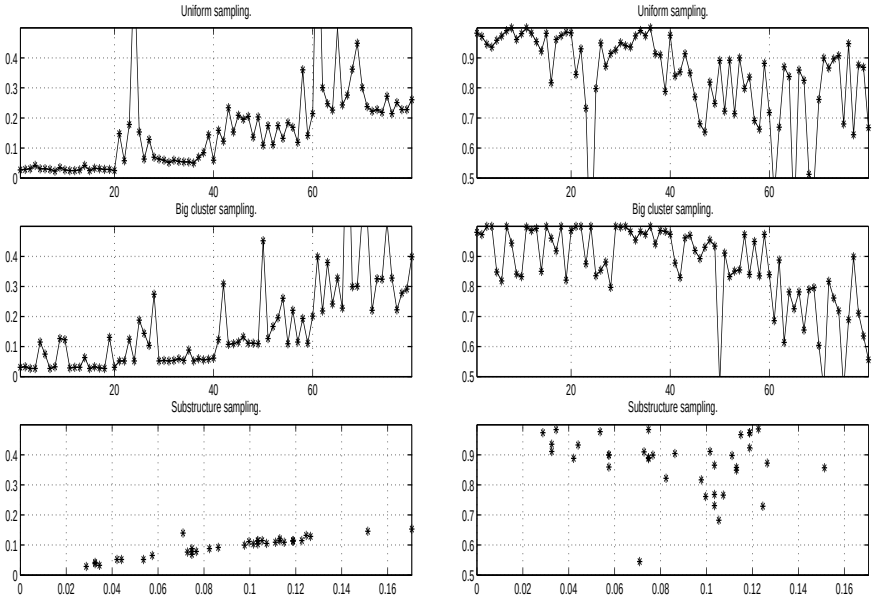


Figure 25: Dendrogram for the `uk` data with $cut < 0.25$. The low cut threshold have been chosen for the visualization purposes.

Cluster ID	Cluster contents
4	uk.sport.football.clubs.rangers, uk.sport.football.clubs.celtic, uk.sport.football.scottish.
2	uk.games.fantasy.football, uk.sport.football, uk.sport.football.american, uk.sport.football.clubs.leeds-united, uk.sport.football.clubs.liverpool, uk.sport.football.clubs.man-city, uk.sport, uk.sport.football.clubs.sunderland, uk.sport.football.clubs.southampton, uk.sport.football.clubs.newcastle-united, uk.sport.football.clubs.west-ham, uk.sport.football.clubs.bradford-city, uk.sport.football.clubs.wimbledon.
3	uk.games.video.dreamcast, uk.games.video.misc, uk.games.video.playstation, uk.games.video.playstation.forsale, uk.games.video.gameboy, uk.games.video.gamecube, uk.games.video.xbox, uk.games.computer.misc.
5	uk.media.tv.friends, uk.media.tv.us-sitcoms.
20	uk.games.trading-cards.misc, uk.games.computer.multiplayer, uk.games.computer.quake, uk.games.computer.quake2, uk.games.computer.quake3, uk.games.misc, uk.games.roleplay, uk.games.trading-cards.marketplace, uk.games, uk.games.board, uk.games.trading-card, uk.games.video, uk.games.computer, uk.games.fantasy, uk.games.computer.counterstrike.
19	uk.org.bcs.announce.
42	uk.media.dvd, uk.media.home-cinema, uk.rec.audio, uk.media.films, uk.media.films.carry-on, uk.media.mags.net, uk.media.mags.uk, uk.media.dvd.cracked.
28	uk.media.tv.cable, uk.tech.digital-tv, uk.tech.broadcast, uk.media.radio, uk.media.tv.sky, uk.tech.tv.sky, uk.lifts, uk.tech.digital-tv.crypt, uk.tech.tv.video.pvr.

Table 13: Selected clusters in the uk dendrogram (from right to left).



(a) Clustering Error.

(b) Correlation between weighted cophenetic matrices.

Figure 26: Stability of the dendrograms for `uk` data. The sample percentages are 0.025, 0.05, 0.10, and 0.20. We have sampled 20 times with each percentage. That is, the points 1–20 represent the samples with sample percentage 0.025, the points 21–40 represent the samples with sample percentage 0.05, and so on. In each subfigure, the topmost plot is for uniform sampling, the middle plot is for big cluster sampling, and the bottom plot is for substructure sampling. In the uniform and big cluster sampling plots, we have the sample number on the x-axis. In the substructure sampling plot, the x-axis represents the sample percentage.

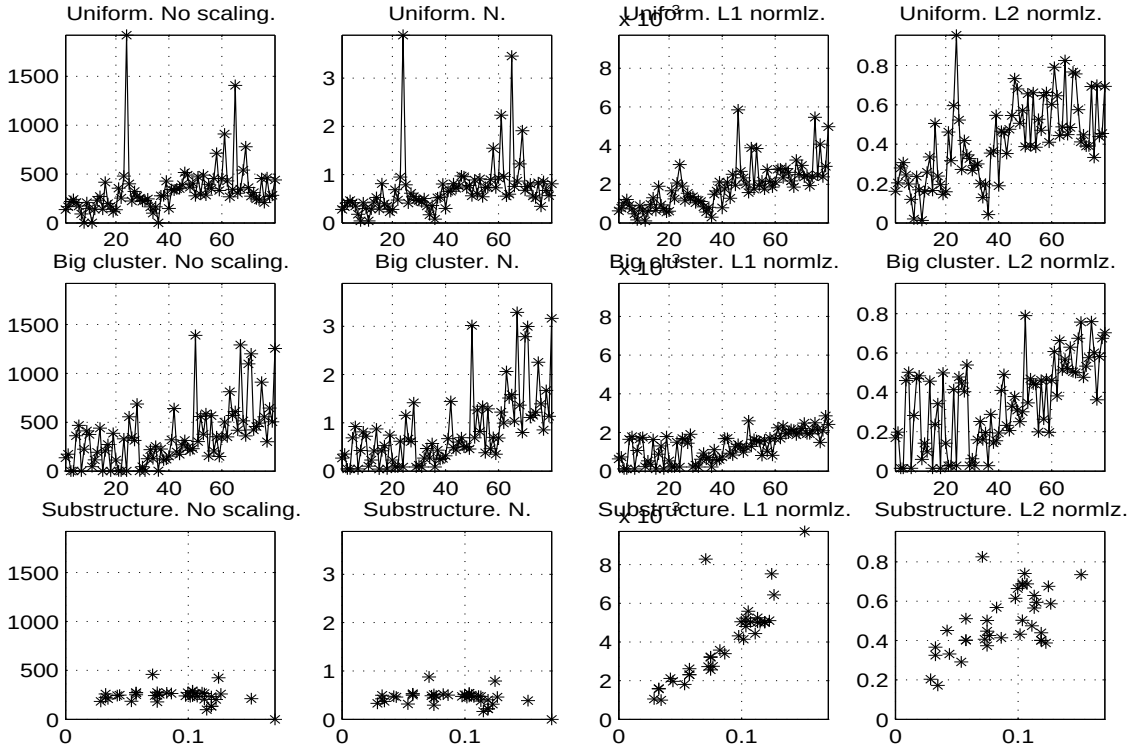
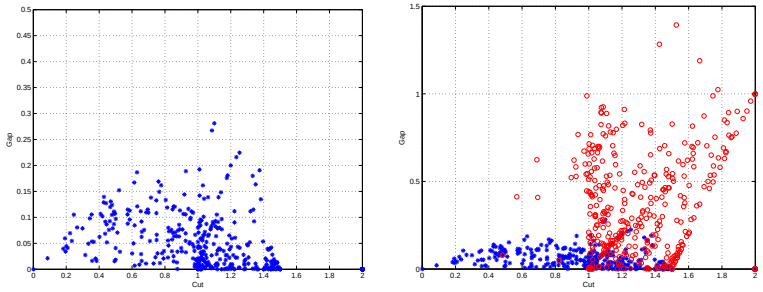


Figure 27: Stability of the dendrograms for `uk` data. L2 distance between weighted cophenetic matrices. The sample percentages are 0.025, 0.05, 0.10, and 0.20. We have sampled 20 times with each percentage. That is, the points 1–20 represent the samples with sample percentage 0.025, the points 21–40 represent the samples with sample percentage 0.05, and so on. Top row: Uniform sampling. Middle row: Big cluster sampling. Bottom row: Substructure sampling. Left column: No scaling of cophenetic matrices. Second column: Cophenetic matrices scaled by N. Third column: Cophenetic matrices with L1 normalization. Right column: Cophenetic matrices with L2 normalization. In the uniform and big cluster sampling plots, we have the sample number on the x-axis. In the substructure sampling plots, the x-axis represents the sample percentage.



(a) All 535 (min cut)-gap pairs.

(b) All 535 (min cut)-gap pairs (blue stars) with random cut-gap pairs (red circles).

Figure 28: Cut vs. gap in the `aus, es` data.

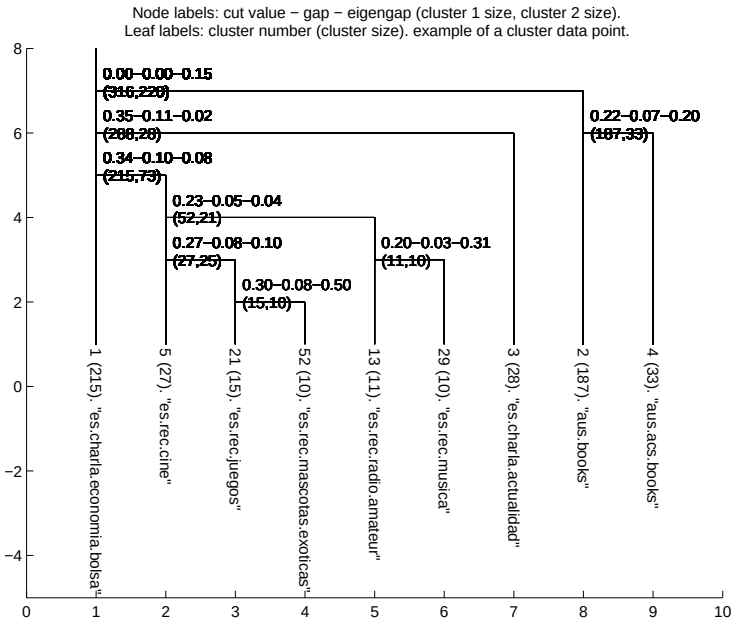
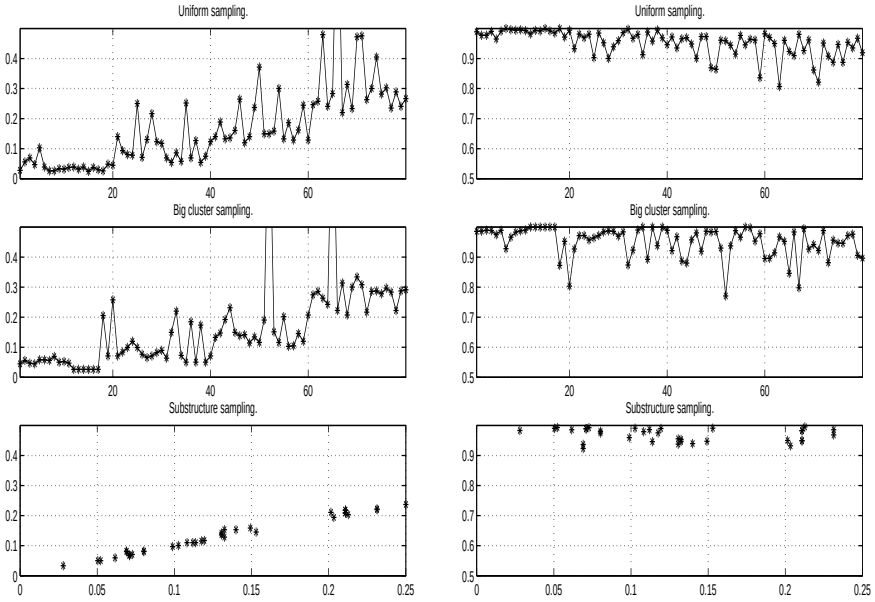


Figure 29: Dendrogram for the `aus, es` data with $cut < 0.4$. The low cut threshold have been chosen for the visualization purposes.

Cluster ID	Cluster contents
29	es.rec.musica, es.rec.musica.alternativas, es.rec.musica.blues, es.rec.musica.grupos.beatles, es.rec.musica.grupos.misc, es.rec.musica.misc, es.rec.musica.techno, es.rec.musica.clasica, es.rec.musica.jazz, es.rec.musica.partituras.
13	es.rec.radio.amateur, es.rec.viajes, es.rec.pasatiempos, es.rec.radio, es.rec.radio.misc, es.rec.radio.ondacorta, es.rec.tv.concursos, es.rec.tv.misc, es.rec.tv.series, es.rec.trenes, es.rec.naturismo.
52	es.rec.mascotas.exoticas, es.rec.misc, es.rec.motor.4x4, es.rec.modelismo, es.rec.mascotas, es.rec.mascotas.gatos, es.rec.mascotas.misc, es.rec.mascotas.peces, es.rec.mascotas.perros, es.rec.labores.
21	es.rec.juegos, es.rec.juegos.comp.arcade, es.rec.juegos.comp.aventuras, es.rec.juegos.comp.misc, es.rec.juegos.comp.simuladores, es.rec.juegos.estrategia, es.rec.manga, es.rec.juegos.misc, es.rec.juegos.rol, es.rec.juegos.ajedrez, es.rec.juegos.comp.simuladores.misc, es.rec.juegos.comp.simuladores.vuelo, es.rec.juegos.magic, es.rec.juegos.pinball, es.eunet.pdsoft.

Table 14: Selected clusters in the `aus,es` dendrogram (from right to left).



(a) Clustering Error.

(b) Correlation between weighted cophenetic matrices.

Figure 30: Stability of the dendrograms for `aus, es` data. The sample percentages are 0.025, 0.05, 0.10, and 0.20. We have sampled 20 times with each percentage. That is, the points 1–20 represent the samples with sample percentage 0.025, the points 21–40 represent the samples with sample percentage 0.05, and so on. In each subfigure, the topmost plot is for uniform sampling, the middle plot is for big cluster sampling, and the bottom plot is for substructure sampling. In the uniform and big cluster sampling plots, we have the sample number on the x-axis. In the substructure sampling plot, the x-axis represents the sample percentage. In substructure sampling, we have exceptionally excluded two clusters, namely clusters 1 and 2. In big cluster sampling, we have first removed points from cluster 1 only (10 first samples for each sample percentage) and then from cluster 2 only (10 last samples for each sample percentage).

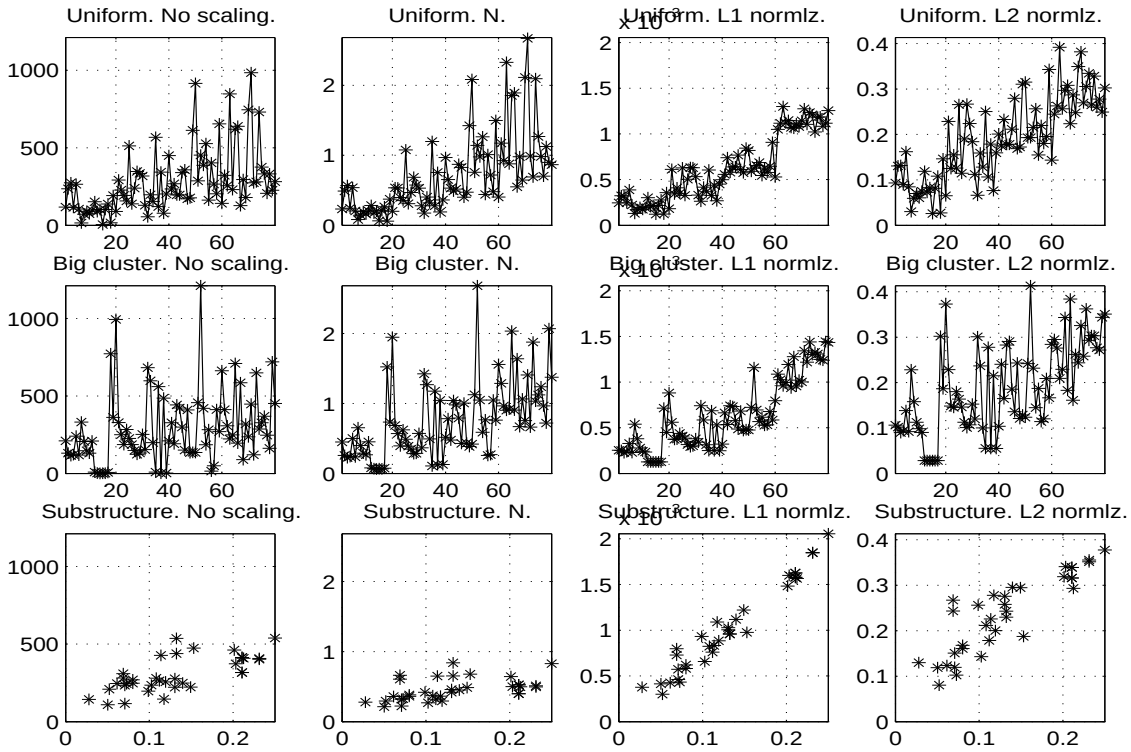
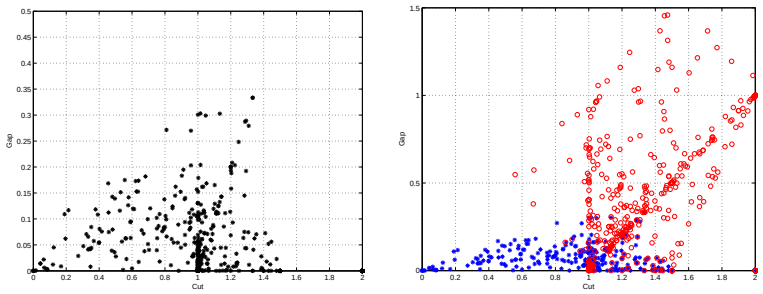


Figure 31: Stability of the dendrograms for `aus, es` data. L2 distance between weighted cophenetic matrices. The sample percentages are 0.025, 0.05, 0.10, and 0.20. We have sampled 20 times with each percentage. That is, the points 1–20 represent the samples with sample percentage 0.025, the points 21–40 represent the samples with sample percentage 0.05, and so on. Top row: Uniform sampling. Middle row: Big cluster sampling. Bottom row: Substructure sampling. Left column: No scaling of cophenetic matrices. Second column: Cophenetic matrices scaled by N. Third column: Cophenetic matrices with L1 normalization. Right column: Cophenetic matrices with L2 normalization. In the uniform and big cluster sampling plots, we have the sample number on the x-axis. In the substructure sampling plots, the x-axis represents the sample percentage. In substructure sampling, we have exceptionally excluded two clusters, namely clusters 1 and 2. In big cluster sampling, we have first removed points from cluster 1 only (10 first samples for each sample percentage) and then from cluster 2 only (10 last samples for each sample percentage).



(a) All 539 (min cut)-gap pairs.

(b) All 539 (min cut)-gap pairs (blue stars) with random cut-gap pairs (red circles).

Figure 32: Cut vs. gap in the `soc, talk` data.

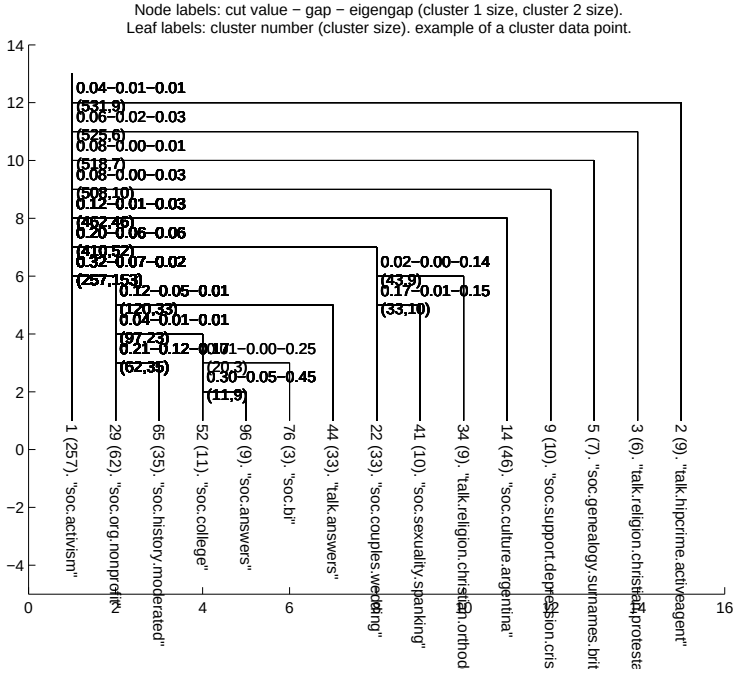
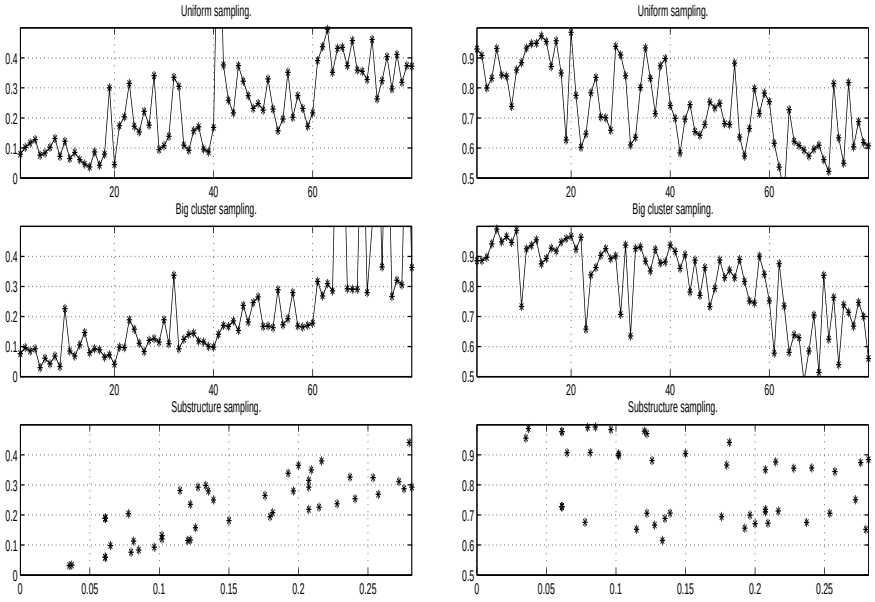


Figure 33: Dendrogram for the `soc, talk` data with $cut < 0.33$. The low cut threshold have been chosen for the visualization purposes.

Cluster ID	Cluster contents
2	talk.hipcrime.activeagent, talk.forgery, talk.gibberish.bill-palmer, soc.hipcrime, talk.hipcrime, talk.bizarre.funny, talk.hipcrime.congressagent, talk.hipcrime.listagent, talk.bizarre.nice.
3	talk.religion.christian.protestant.baptist, talk.religion.christian.protestant.charismatic, talk.religion.christian.protestant.evangelical, talk.religion.christian.protestant, talk.religion.christian.protestant.episcopal, talk.religion.christian.protestant.adventist.
5	soc.genealogy.surnames.britain, soc.genealogy.surnames.canada, soc.genealogy.surnames.global, soc.genealogy.surnames.misc, soc.genealogy.surnames.usa, soc.genealogy.surnames.german, soc.genealogy.surnames.ireland.
9	soc.support.depression.crisis, soc.support.abuse.sexual, soc.support.depression.family, soc.support.depression.manic, soc.support.depression.seasonal, soc.support.depression.treatment, soc.support.depression.misc, soc.support.depression, soc.support.depression.crises, soc.support.youth.
34	talk.religion.christian.orthodox.greek, talk.religion.christian.orthodox, talk.religion.christian.jehovah-witness, talk.religion.christian.apostolic, talk.religion.christian.arian, talk.religion.christian.coptic, talk.religion.christian.mormon, talk.religion.christian.orthodox.misc, talk.religion.christian.nestorian.
41	soc.sexuality.spanking, soc.subculture.bondage-bdsm.femdom, soc.subculture.bondage-bdsm, soc.sexuality.perversions.sodomy, soc.sexuality.shrimping, soc.subculture.bondage-bdsmuw, soc.subculture.homosexual, soc.support.youth.gay-lesbian-bi, soc.subculture, soc.rights.alien.
76	soc.bi, soc.motss, soc.support.transgendered.
96	soc.answers, soc.culture.jewish.parenting, soc.feminism, soc.support.loneliness, soc.history.war.us-civil-war, soc.org.freemasonry, soc.support.pregnancy.loss, soc.culture.jewish.moderated, soc.culture.jewish.holocaust.
52	soc.college, soc.misc, soc.college.admissions, soc.college.financial-aid, soc.college.grad, soc.college.gradinfo, soc.college.org.aiesec, soc.college.teaching-asst, soc.college.graduation, soc.college.org, soc.cultural.israel.

Table 15: Selected clusters in the `soc, talk` dendrogram (from right to left).



(a) Clustering Error.

(b) Correlation between weighted cophenetic matrices.

Figure 34: Stability of the dendrograms for `soc, talk` data. The sample percentages are 0.025, 0.05, 0.10, and 0.20. We have sampled 20 times with each percentage. That is, the points 1–20 represent the samples with sample percentage 0.025, the points 21–40 represent the samples with sample percentage 0.05, and so on. In each subfigure, the topmost plot is for uniform sampling, the middle plot is for big cluster sampling, and the bottom plot is for substructure sampling. In the uniform and big cluster sampling plots, we have the sample number on the x-axis. In the substructure sampling plot, the x-axis represents the sample percentage.

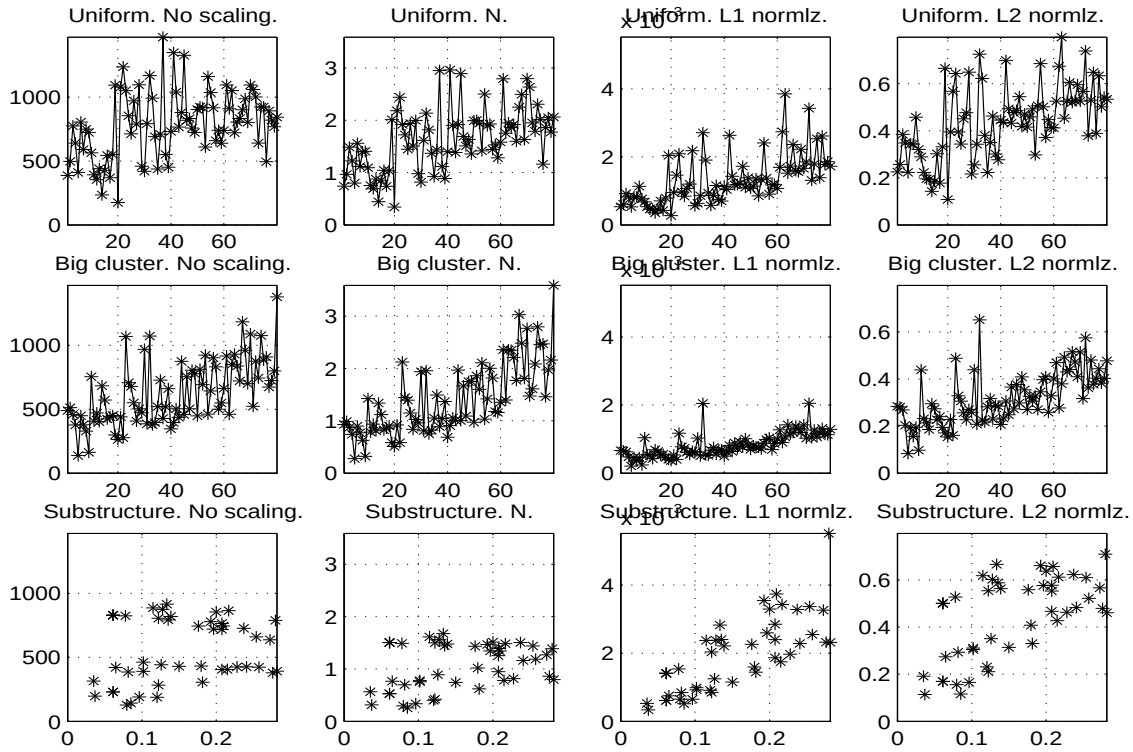
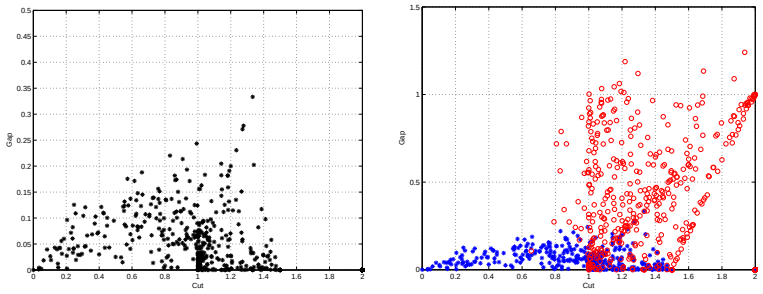


Figure 35: Stability of the dendrograms for `soc, talk` data. L2 distance between weighted cophenetic matrices. The sample percentages are 0.025, 0.05, 0.10, and 0.20. We have sampled 20 times with each percentage. That is, the points 1–20 represent the samples with sample percentage 0.025, the points 21–40 represent the samples with sample percentage 0.05, and so on. Top row: Uniform sampling. Middle row: Big cluster sampling. Bottom row: Substructure sampling. Left column: No scaling of cophenetic matrices. Second column: Cophenetic matrices scaled by N. Third column: Cophenetic matrices with L1 normalization. Right column: Cophenetic matrices with L2 normalization. In the uniform and big cluster sampling plots, we have the sample number on the x-axis. In the substructure sampling plots, the x-axis represents the sample percentage.



(a) All 619 (min cut)-gap pairs.

(b) All 619 (min cut)-gap pairs (blue stars) with random cut-gap pairs (red circles).

Figure 36: Cut vs. gap in the `uk, us` data.

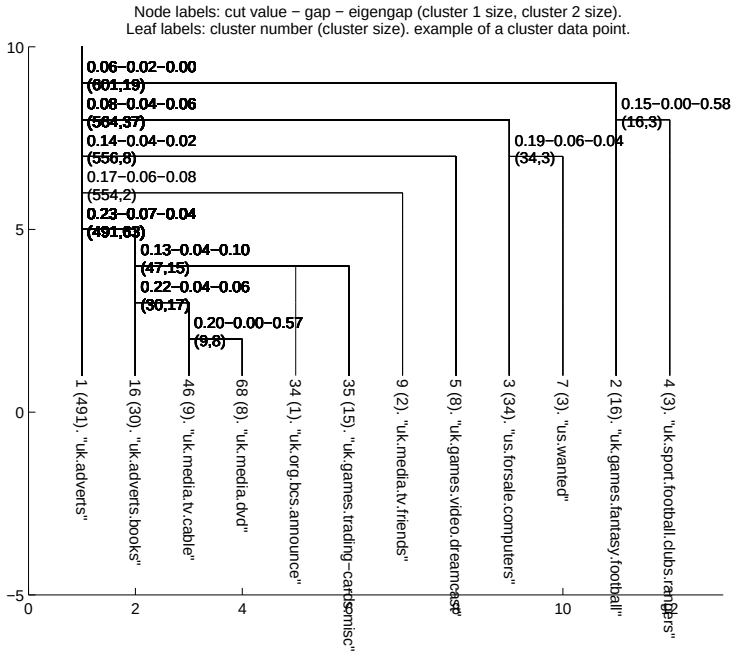
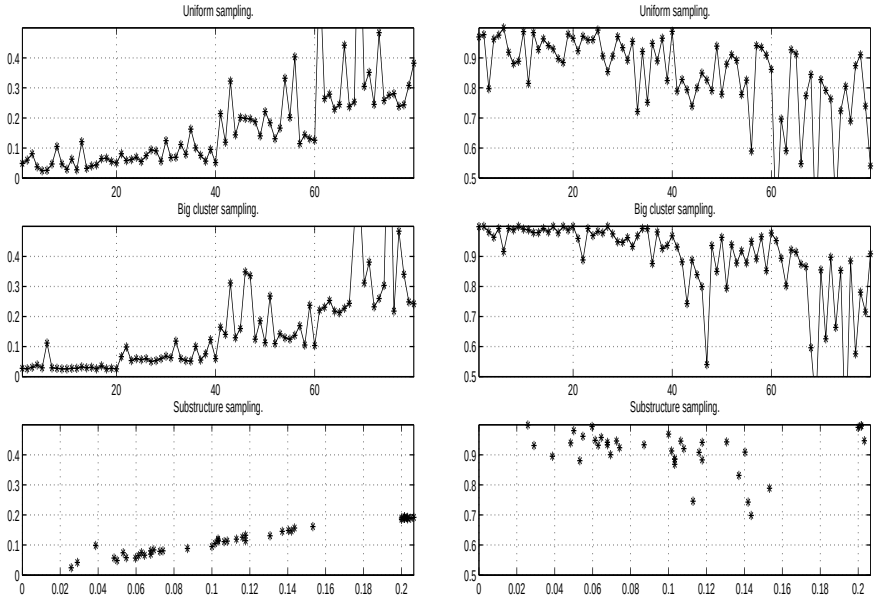


Figure 37: Dendrogram for the `uk, us` data with $cut < 0.25$. The low cut threshold have been chosen for the visualization purposes.

Cluster ID	Cluster contents
4	uk.sport.football.clubs.rangers, uk.sport.football.clubs.celtic, uk.sport.football.scottish.
2	uk.games.fantasy.football, uk.sport.football, uk.sport.football.american, uk.sport.football.clubs.leeds-united, uk.sport.football.clubs.liverpool, uk.sport.football.clubs.man-city, us.sport.football.misc, uk.sport, uk.sport.football.clubs.sunderland, uk.sport.football.clubs.southampton, us.sport.football.pro, us.sport.football, uk.sport.football.clubs.newcastle-united, uk.sport.football.clubs.west-ham, uk.sport.football.clubs.bradford-city, uk.sport.football.clubs.wimbledon.
7	us.wanted, us.wanted.misc, us.wanted.d.
5	uk.games.video.dreamcast, uk.games.video.misc, uk.games.video.playstation, uk.games.video.playstation.forsale, uk.games.video.gameboy, uk.games.video.gamecube, uk.games.video.xbox, uk.games.computer.misc.
9	uk.media.tv.friends, uk.media.tv.us-sitcoms.
35	uk.games.trading-cards.misc, uk.games.computer.multiplayer, uk.games.computer.quake, uk.games.computer.quake2, uk.games.computer.quake3, uk.games.misc, uk.games.roleplay, uk.games.trading-cards.marketplace, uk.games, uk.games.board, uk.games.trading-card, uk.games.video, uk.games.computer, uk.games.fantasy, uk.games.computer.counterstrike.
34	uk.org.bcs.announce.
68	uk.media.dvd, uk.media.home-cinema, uk.rec.audio, uk.media.films, uk.media.films.carry-on, uk.media.mags.net, uk.media.mags.uk, uk.media.dvd.cracked.
46	uk.media.tv.cable, uk.tech.digital-tv, uk.tech.broadcast, uk.media.radio, uk.media.tv.sky, uk.tech.tv.sky, uk.lifts, uk.tech.digital-tv.crypt, uk.tech.tv.video.pvr.

Table 16: Selected clusters in the `uk, us` dendrogram (from right to left).



(a) Clustering Error.

(b) Correlation between weighted cophenetic matrices.

Figure 38: Stability of the dendrograms for `uk, us` data. The sample percentages are 0.025, 0.05, 0.10, and 0.20. We have sampled 20 times with each percentage. That is, the points 1–20 represent the samples with sample percentage 0.025, the points 21–40 represent the samples with sample percentage 0.05, and so on. In each subfigure, the topmost plot is for uniform sampling, the middle plot is for big cluster sampling, and the bottom plot is for substructure sampling. In the uniform and big cluster sampling plots, we have the sample number on the x-axis. In the substructure sampling plot, the x-axis represents the sample percentage.

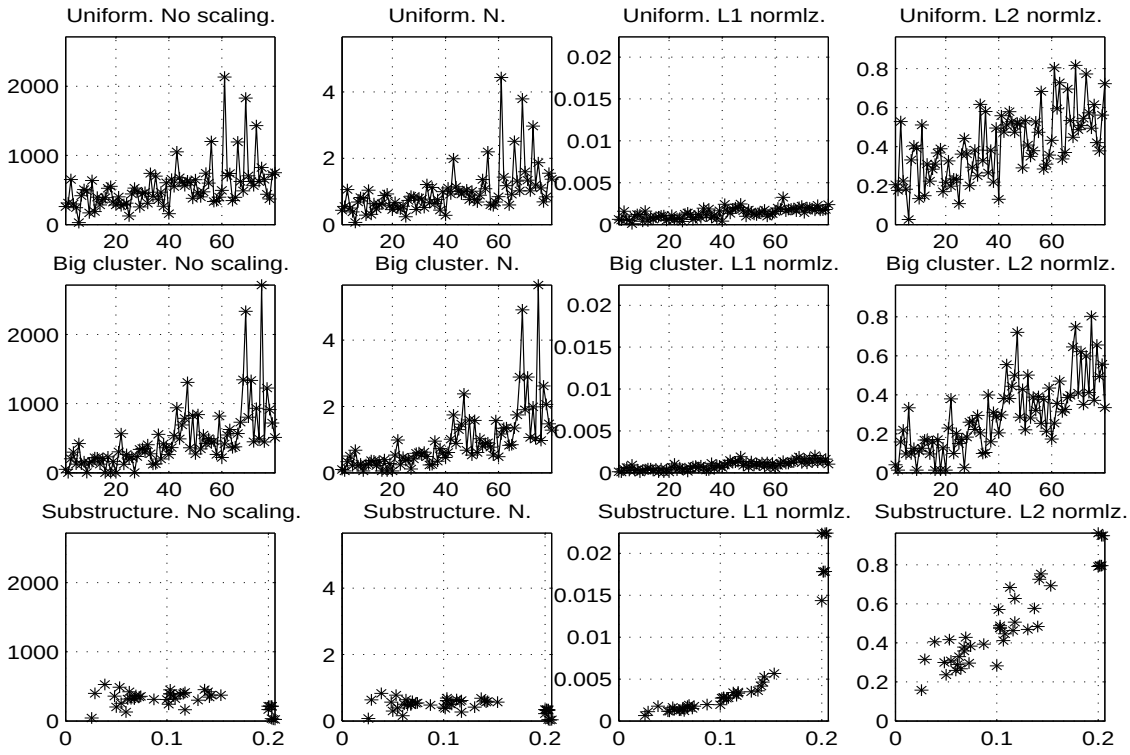
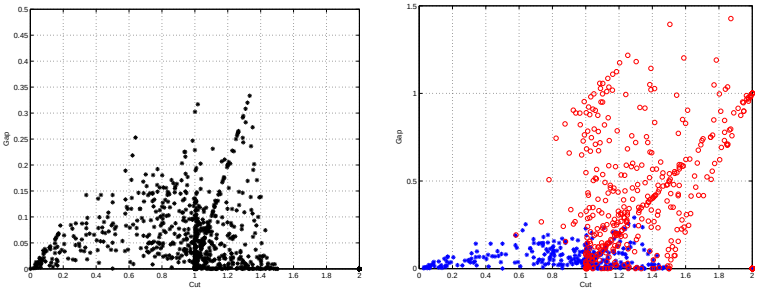


Figure 39: Stability of the dendrograms for `uk, us` data. L2 distance between weighted cophenetic matrices. The sample percentages are 0.025, 0.05, 0.10, and 0.20. We have sampled 20 times with each percentage. That is, the points 1–20 represent the samples with sample percentage 0.025, the points 21–40 represent the samples with sample percentage 0.05, and so on. Top row: Uniform sampling. Middle row: Big cluster sampling. Bottom row: Substructure sampling. Left column: No scaling of cophenetic matrices. Second column: Cophenetic matrices scaled by N. Third column: Cophenetic matrices with L1 normalization. Right column: Cophenetic matrices with L2 normalization. In the uniform and big cluster sampling plots, we have the sample number on the x-axis. In the substructure sampling plots, the x-axis represents the sample percentage.



(a) Sample of (min cut)-gap pairs.

(b) Sample of (min cut)-gap pairs (blue stars) with random cut-gap pairs (red circles).

Figure 40: Cut vs. gap in the `rec` data.

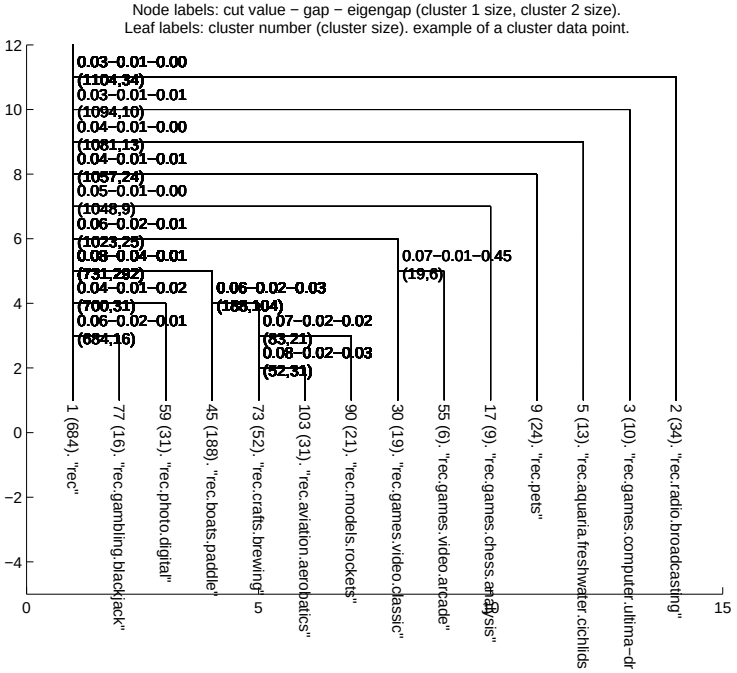
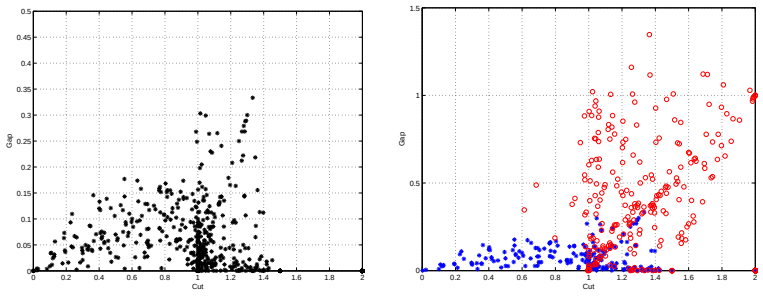


Figure 41: Dendrogram for the `rec` data with $cut < 0.08$. The low cut threshold have been chosen for the visualization purposes.

Cluster ID	Cluster contents
3	rec.games.computer.ultima-dragons, rec.games.computer.ultima.dragons, rec.games.computer.ultima.series, rec.games.computer.stars, rec.games.computer.ultima.online, rec.games.computer.xpilot, rec.games.computer.ultima, rec.games.computer.ultima-online, rec.games.computer.ultima.misc, rec.games.cyber.
5	rec.aquaria.freshwater.cichlids, rec.aquaria.freshwater.goldfish, rec.aquaria.freshwater.misc, rec.aquaria.freshwater.plants, rec.aquaria.marine.misc, rec.aquaria.marine.reefs, rec.aquaria.marketplace, rec.aquaria.misc, rec.aquaria.tech, rec.aquaria, rec.aquaria.freshwater, rec.aquaria.marine, rec.aquaria.freshwater.fish.
17	rec.games.chess.analysis, rec.games.chess.computer, rec.games.chess.misc, rec.games.chess.play-by-email, rec.games.chess, rec.games.chess.politics, rec.games.chinese-chess, rec.games.go, rec.games.chess.computers.
55	rec.games.video.arcade, rec.games.video, rec.games.video.arcade.collecting, rec.games.video.arcade.marketplace, rec.games.computer, rec.games.pinball.
30	rec.games.video.classic, rec.games.video.misc, rec.games.video.sony, rec.games.video.marketplace, rec.games.video.nintendo, rec.games.video.sega, rec.arts.sf.starwars.vs.startrek, rec.games.video.advocacy, rec.games.computer.everquest, rec.games.video.nintendo.n64, rec.games.video.sony-playstation, rec.games.video.atari, rec.games.video.3do, rec.games.vectrex, rec.games.video.cd-i, rec.games.video.cd32, rec.games.video.colecovision, rec.games.video.intellivision, rec.games.xtank.play.
77	rec.gambling.blackjack, rec.gambling.craps, rec.gambling.lottery, rec.gambling.misc, rec.gambling.other-games, rec.gambling.poker, rec.gambling.racing, rec.gambling.sports, rec.gambling, rec.gambling.blackjack.moderated, rec.gambling.games, rec.gambling.lotto, rec.gambling.small-stakes, rec.sport.cricket-racing, rec.travel.australia, rec.forsale.computers.pc-clone.

Table 17: Selected clusters in the `rec` dendrogram (from right to left).



(a) Sample of (min cut)-gap pairs.

(b) Sample of (min cut)-gap pairs (blue stars) with random cut-gap pairs (red circles).

Figure 42: Cut vs. gap in the `comp, sfnet, soc` data.

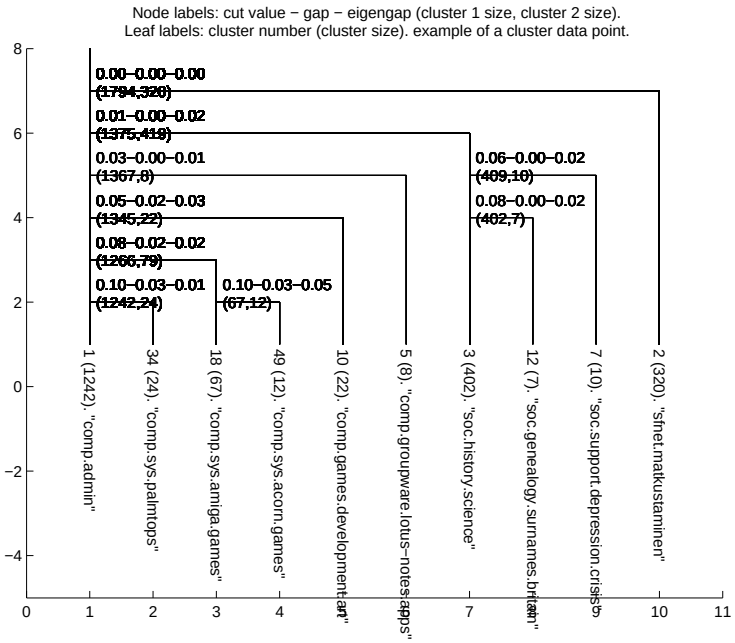
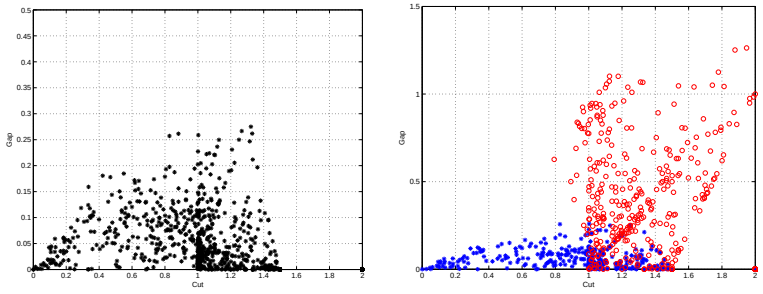


Figure 43: Dendrogram for the `comp, sfnet, soc` data with $cut < 0.1$. The low cut threshold have been chosen for the visualization purposes.

Cluster ID	Cluster contents
7	soc.support.depression.crisis, soc.support.abuse.sexual, soc.support.depression.family, soc.support.depression.manic, soc.support.depression.seasonal, soc.support.depression.treatment, soc.support.depression.misc, soc.support.depression, soc.support.depression.crises, soc.support.youth.
12	soc.genealogy.surnames.britain, soc.genealogy.surnames.canada, soc.genealogy.surnames.global, soc.genealogy.surnames.misc, soc.genealogy.surnames.usa, soc.genealogy.surnames.german, soc.genealogy.surnames.ireland.
5	comp.groupware.lotus-notes.apps, comp.groupware.lotus-notes.programmer, comp.groupware.groupwise, comp.groupware.lotus-notes, comp.groupware.lotus-notes.admin, comp.groupware.lotus-notes.misc, comp.groupware.lotus-notes.programm, comp.home.
49	comp.sys.acorn.games, comp.sys.acorn.announce, comp.sys.acorn.extra-cpu, comp.sys.acorn.advocacy, comp.sys.acorn.hardware, comp.sys.acorn.networking, comp.sys.acorn.programmer, comp.sys.acorn.apps, comp.sys.acorn.misc, comp.sys.acorn.tech, comp.sys.alliant, comp.sys.acorn.

Table 18: Selected clusters in the `comp, sfnet, soc` dendrogram (from right to left).



(a) Sample of (min cut)-gap pairs.

(b) Sample of (min cut)-gap pairs (blue stars) with random cut-gap pairs (red circles).

Figure 44: Cut vs. gap in the `microsoft` data.

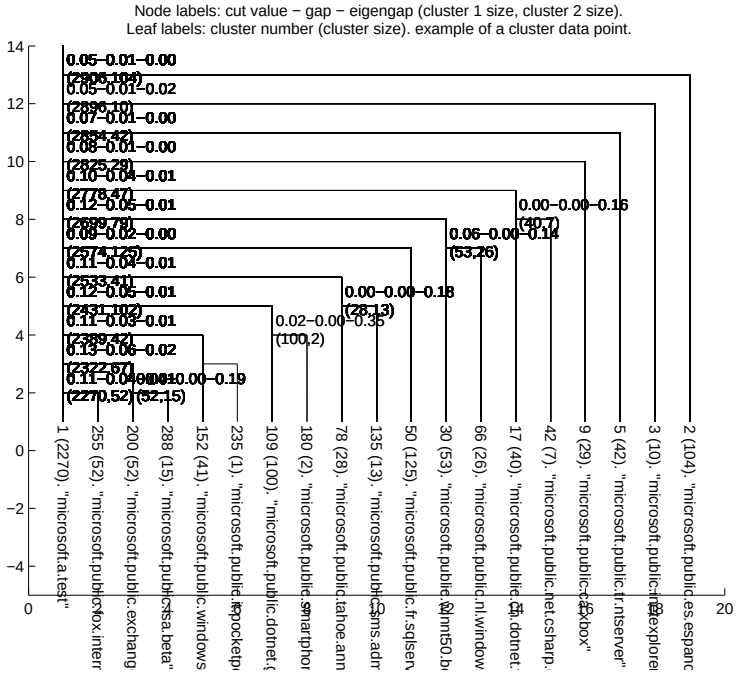


Figure 45: Dendrogram for the `microsoft` data with $cut < 0.15$. The low cut threshold have been chosen for the visualization purposes.

Cluster ID	Cluster contents
3	microsoft.public.inetexplorer.ie4.outlookexpress.stationery, microsoft.public.windows.inetexplorer.ie5.outlookexpress.stationery, microsoft.public.windows.inetexplorer.ie55.outlookexpress.stationery, microsoft.public.se.design.gallery, microsoft.public.br.design.gallery, microsoft.public.espanol.design.gallery, microsoft.public.it.design.gallery, microsoft.public.it.dotnet.beta.uddi, microsoft.public.windows.inetexplorer.ie6_outlookexpress.stationery, microsoft.public.it.dotnet.uddi.
42	microsoft.public.net.csharp.general, microsoft.public.net.general, microsoft.public.net.framework.aspplus.general, microsoft.public.net.framework.sdk.setup, microsoft.public.net.framework.classes.general, microsoft.public.net.framework.runtime.general, microsoft.public.net.framework.sdk.general.
135	microsoft.public.sms.admin, microsoft.public.sms.installer, microsoft.public.sms.inventory, microsoft.public.sms.misc, microsoft.public.sms.netmon, microsoft.public.sms.rcdiags, microsoft.public.sms.setup, microsoft.public.sms.sharedapps, microsoft.public.sms.swdist, microsoft.public.sms.sitecomm, microsoft.public.sms.tools, microsoft.public.sms, microsoft.public.sms.swmeter.
180	microsoft.public.smartphone.developer, microsoft.public.smartphone.
235	microsoft.public.it.pocketpc.marketplace.
288	microsoft.public.isa.beta, microsoft.public.isa, microsoft.public.isa.enterprise, microsoft.public.isaserver, microsoft.public.de.german.isaserver, microsoft.public.jp.isa, microsoft.public.isa.edu, microsoft.public.isa.sdk-dev, microsoft.public.isa.configuration, microsoft.public.cn.isaserver, microsoft.public.isa.publishing, microsoft.public.isa.clients, microsoft.public.isa.vpn, microsoft.public.isa.wishlist, microsoft.public.de.german.windows.terminaldienste.

Table 19: Selected clusters in the `microsoft` dendrogram (from right to left).

9 Conclusion

We have explored the community structure of Usenet newsgroups using a part of the Microsoft Netscan data. Our data set consists of crossposting information for 89,687 newsgroups over a period of 3.4 years, with a total number of crosspostings exceeding 700 million. Because of the large size of the data set, we have not been able to analyze the whole data at once. Instead, we have taken a closer look at ten subsets of the data, namely the `us`; `talk`; `religion`; `uk`; `aus, es`; `soc, talk`; `uk, us`; `rec`; `soc, comp, sfnet`; and `microsoft` hierarchies.

We have presented different approaches to explore the community structure of the newsgroup data: spectral clustering, spectral hierarchical clustering, and consensus clustering with mean connectivity matrices. Further, we have discussed in detail various ways to evaluate the quality of spectral clustering and identified 72 different approaches. We have applied all these data processing methods to the `talk` data and analyzed the results comprehensively. Since the topic of spectral hierarchical clusterings is most novel and most promising in our setting, we have focused solely on spectral hierarchical clusterings with the remaining 9 data sets.

Spectral hierarchical clustering has been presented implicitly in [12] and addressed explicitly but briefly in [3]. Flat spectral clustering and consensus clustering are well known data analysis methods [14, 10, 12, 18, 19, 17, 9]. Evaluating the quality of a hierarchical clustering is a new topic; a related topic of comparing hierarchical clusterings has been addressed in [5].

Thus, the main contributions of this report are a) studying the topic of spectral hierarchical clustering in detail, including novel aspects such as pruning the dendrogram; b) presenting the first study on evaluating the quality of spectral hierarchical clusterings; c) investigating the community structure of the Usenet with the help of various data analysis methods.

Acknowledgements

This work has been supported by University of Washington Center for Statistics and the Social Sciences (SCCC) Seed Grant. The authors are grateful to James Kitts from the University of Washington, Marc Smith from Microsoft Research, and the Microsoft Netscan team for fruitful collaboration.

References

[1] Netscan data analysis results. `http://www.ms.washington.edu/spectral/netscan/`.

[2] What is Usenet? Frequently Asked Questions. `http://www.faws.org/faws/usenet/what-is/part1/`.

[3] C. Borgs, J. Chayes, M. Mahdian, and A. Saberi. Exploring the community structure of newsgroups. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2004.

[4] G. Flake, S. Lawrence, and C. L. Giles. Efficient identification of web communities. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000.

[5] E. B. Fowlkes and C. L. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78(383):553–569, 1983.

[6] J. Hopcroft, O. Khan, B. Kulis, and B. Selman. Natural communities in large linked networks. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

[7] J. Hopcroft, O. Khan, and B. Selman. Tracking evolving communities in large linked networks. *Proceedings of the National Academy of Sciences, Supplement 1*, 101:5249–5253, 2004.

[8] M. Meila. Comparing clusterings by the variation of information. In *Proceedings of the Sixteenth Annual Conference on Learning Theory*, 2003.

[9] M. Meila and L. Xu. Multiway cuts and spectral clustering. Technical Report 442, University of Washington, May 2003. `www.stat.washington.edu/mmp/Papers/nips03-multicut-tr.ps`.

[10] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, 2002.

[11] A. Patrikainen and M. Meila. Comparing subspace clusterings. Technical report, University of Washington, 2004.

[12] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2000.

[13] Marc A. Smith. Invisible crowds in cyberspace: Mapping the social structure of the usenet. In *Communities in Cyberspace*, chapter 8, pages 195–219. Routledge, 1999.

[14] A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research (JMLR)*, 3:583–617, 2002.

[15] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Academic Press, 1999.

[16] M. Toyoda and M. Kitsuregawa. Extracting evolution of web communities from a series of web archives. In *Proceedings of the 14th ACM Conference on Hypertext and Hypermedia*, 2003.

[17] D. Verma and M. Meila. A comparison of spectral clustering algorithms. Technical report, University of Washington, 2003.

[18] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *International Conference on Computer Vision*, 1999.

[19] S. X. Yu and J. Shi. Multiclass spectral clustering. In *International Conference on Computer Vision*, 2003.