Computational Proxies of Large Inputs: Improving Efficiency through Sampling

by

Christopher L. Bennett

(Under the direction of Dr. Walter D. Potter)

Abstract

Complete processing can rapidly become intractable for extremely large inputs to computational tasks, and sampling is often employed to improve efficiency. In many such applications, sampling is used to estimate features of a population by enumerating features for a subset of that population. This work introduces the idea of sampling for large input proxies (SLIP) for use in problems where solutions depend on problem specific features (PSF). Random sampling schemes are introduced which retain such features for several applications, and approximations of the original input's features consequently can take significantly less time. In addition, these proxies can actually be used as substitute inputs for more efficient computation for some tasks. SLIP is applied to two sorting related problems as well as two data mining tasks (classification and clustering). With different types of inputs from different domains, SLIP provides a general framework to improve performance in several applications with little loss in accuracy.

Index words: Sampling, Sorting Algorithm Selection, Graph Classification, Graph Clustering

Computational Proxies of Large Inputs: Improving Efficiency through Sampling

by

Christopher L. Bennett

B.B.A., The University of Georgia, 1993

A Dissertation Submitted to the Graduate Faculty
of The University of Georgia in Partial Fulfillment
of the
Requirements for the Degree

Doctor of Philosophy

Athens, Georgia

2007

Computational Proxies of Large Inputs:
Improving Efficiency through Sampling

by

Christopher L. Bennett

Approved:

Major Professor: Dr. Walter D. Potter

Committee: Dr. Lynne Billard
Dr. E. Rodney Canfield
Dr. Shelby Funk
Dr. Richard Watson

Electronic Version Approved:

Dr. Maureen Grasso
Dean of the Graduate School
The University of Georgia
May 2007

Acknowledgments

I would like to acknowledge the help and support of my advisor, Dr. Walter D. Potter. He was always open to new ideas, and his excitement and dedication were always infectious. To the rest of my committee, your support and advice were invaluable. I would also like to thank the University of Georgia and its Department of Computer Science for providing support throughout my graduate career.

Finally, I would like to acknowledge the love and support of my parents, Vickie and Joe, and of my wife, Shannon. Thanks for everything.

Table of Contents

Page

Acknowledgments iv

List of Figures vii

List of Tables x

Chapter

1 Introduction 1

1.1 Sampling and Problem Specific Features 2

1.2 Applications 5

1.3 Organization 8

2 Survey of Related Literature 10

2.1 Sampling in Computation 10

2.2 Input Features 15

2.3 Relationship with this work 17

3 Sorting Algorithm Selection 18

3.1 Introduction 18

3.2 Approximating Features of Sortedness using SLIP 22

3.3 Sorting Algorithm Selection 32

3.4 Selecting a Shell sort Increment Set 41

3.5 Conclusions 45

4 Classifying Large Graphs of Newsgroup Data 47

4.1 Introduction 48

4.2 Related Work 49
4.3 Sampling and Graphs 51
4.4 Graph Similarity Measures 53
4.5 Sampling for Similarity 59
4.6 Classification with Samples 84
4.7 Conclusions 95

5 Clustering Large Graphs 96
5.1 Introduction 96
5.2 Related Work 97
5.3 Data Sets 99
5.4 Sampling Methods 101
5.5 Distance Metrics 102
5.6 Clustering Experiments 105
5.7 Results 107
5.8 Conclusions 112

6 Conclusion 113
6.1 Summary 113
6.2 Future Work 115

Bibliography 118

Appendix

A Newsgroups Graph Distance Data 126

B Yahoo News! Articles Graph Distance Data 151

List of Figures

3.1 A comparison of Quick and Shell sorts on different types of inputs. The performance of Shell sort severely degrades when more than 1,000 random element exchanges are performed. 21
3.2 Example of in-order sampling method for a permutation. 25
3.3 A comparison of sorting algorithm performance with 5,000,000 elements. For each type of input (1, 2, 4, or 8 runs), Shell sort degrades in performance from best to worst. The other algorithms remain fairly steady across all ranges of sortedness with some slight decrease in performance as the input becomes random. 34
3.4 A comparison of quick sort, merge sort, Shell sort, and heap sort on permutations of size 5,000,000 with 1, 2, 4, and 8 runs at a 5% sample rate. The same pattern as in Figure 3.3.2 emerges. That is, Shell sort degrades as the various inputs become less and less sorted. The other relative rankings are similar as well as we see quick sort become the best performer while heap sort remains the worst performer. 36
3.5 A comparison of quick sort, merge sort, Shell sort, and heap sort on permutations of size 5,000,000 with 1, 2, 4, and 8 runs at a 10% sample rate The performance of the algorithms match even more closely to the performance on the original inputs as shown in Figure 3.3.2 with the larger proxy size. 37

3.6 A comparison of quick sort, merge sort, Shell sort, and heap sort on permutations of size 5,000,000 with 1, 2, 4, and 8 runs at a 25% sample rate. Again, we see that the larger proxies more accurately reflect the performance of the algorithms on the original inputs. . 38
3.7 A comparison of performance on one run permutations at 5% sample rate. 39
3.8 A comparison of performance on one run permutations at 10% sample rate. 40
3.9 A comparison of performance on one run permutations at 25% sample rate. 40
3.10 Summary of data from various Shell sort increment sets. As the inputs become more random, we see the Knuth and Sedgewick increment sets begin to outperform the Shell and Gonnet sets. With 2 runs in the input, we see that the Shell and Gonnet increment sets quickly become very bad solutions compared to other sets. The performance on the proxies reflect this. 44
4.1 Flexibility of the Hybrid Graph Sampling Technique to Preserve Structure. 61
4.2 Relative changes in distance rankings from original to 10% samples. Node-edge more accurately retains relative distance rankings in its proxies. 81
4.3 Relative changes in distance rankings from original to 30% samples. Again we see that node-edge more accurately retains relative distance rankings in its proxies. The exception is the UGU metric which still does quite well with the random walk proxy. 82

4.4 Relative changes in distance rankings from original to 50% samples. Cutting the graph's size in half produces very little change the relative rankings in distance from one graph to another. 83

5.1 Changes in relative distance from original to 50% proxy. Overall, the maximum common subgraph metric appears to maintain ordering the best for the relatively small graphs created from the news articles. We see that SID begins to break down in its performance as the inputs become smaller. 104

List of Tables

3.1 Comparison of Sortedness Features for Inputs and Samples. Each block represents an input with 10^i random perturbations as well as the averages of 30 proxies of size 1, 5, and 10%. For the inputs and proxies, the features of sortedness are compared both in the absolute and in the relative rates. Because all but the number of inversions is bounded by the size of the input (or proxy), the rates provide standardized measures. For inversions, though, the figures must still be scaled by a factor of the sample rate for a more standardized comparison. We see that as expected larger proxies provide more accurate approximations, but even very small proxies with respect to the size of the input provide accurate notions of the features of sortedness 29

3.2 Performance Summary for Sorting Algorithms 33

3.3 Original versus Sample Sorting Performance. 5,000,000 elements, one run. As the inputs become more random, we see that the quicksort and shellsort begin to swap place as the most efficient. The emphasized times show the areas where for inputs with one original run, the proxy is a bit deceptive with respect to the actual performance of the shellsort. 35

4.1 Categories of Newsgroups from UC-Irvine Repository 63

4.2 Distance between original input using maximum common subgraph. The closest graph for each is row is in bold. 65

4.3 Distance between original Input using WGU metric. The relative distances are very similar to the maximum common subgraph distances. 66
4.4 Distance between original input using UGU metric. UGU values are non-normalized – a fact that will play an important part in its use in the classification task. 67
4.5 Distance between original input using Minimum Common Supergraph. 69
4.6 Distance between original input using Sample Intersection metric. The SID distance measure appears to do an excellent job of approximating the maximum common subgraph metric. 71
4.7 mcs Distance between Node-Edge Samples (15%). Although the absolute values are different than those for the mcs metric for the original graphs, the relative values are similar. 74
4.8 mcs Distance between Node-Edge Samples (30%). For this larger sample size, the relative mcs distance ordering between each graph is approximated very well by the proxy. 75
4.9 mcs Distance between Node-Edge Samples (50%). For the largest sample size, we see that the absolute distances are a little closer but the relative rankings of the distances between graphs is a highly accurate approximation. 76
4.10 mcs Distance between Random Walk Samples (15%). The smallest proxy created using random walk sampling does a good job of approximating the relative ordering of distances between the newsgroups, but we will see that generally the node-edge sampling technique outperforms it in accuracy. 77

4.11 mcs Distance between Random Walk Samples (30%). As one would expect, the random walk samples become more accurate in their approximations with larger sample sizes. 78
4.12 mcs Distance between Random Walk Samples (50%). The largest sample size using a random walk does the best job of approximating the mcs distances between the graphs based on the original inputs. 79
4.13 Classification of Newsgroups by Max Common Subgraph with $K = 3$ and $K = 5$ 90
4.14 Classification of Newsgroups by WGU with $K = 3$ and $K = 5$ 91
4.15 Classification of Newsgroups by UGU with $K = 3$ and $K = 5$ 92
4.16 Classification of Newsgroups by Min Common Supergraph with $K = 3$ and $K = 5$ 93
4.17 Classification of Newsgroups by SID with k=3,k=5 94
5.1 News Article Categories from Yahoo! News 99
5.2 The 41 news subcategories grouped into the seven general categories from Yahoo! News. The inputs will be clustered into seven groups in order to see how well they recreated the Yahoo! assigned clusterings. 100
5.3 Changes in Relative Distance Ranking from Original to Proxy 103
5.4 Cluster performance indexes for Yahoo! News graphs and their proxies 108
5.5 Clusterings produced with global *k*-means on original inputs. These are the clusterings produced from the original inputs and consequently the ones we hope to emulate with the proxy clusterings. The medians for each cluster are listed, and the Rand and Dunn indexes for each distance metric are also provided. 109

5.6 Clusterings produced with global k-means on 50% samples. For each distance metric, the contents of all seven clusters are provided as well as the Rand and Dunn indexes. The proxies are clustered very similarly to the originals, and the indexes indicate they do an excellent job while using significantly less space. 110
5.7 Closest and Most Distant Original Graphs Used in Dunn Index . . 111
5.8 Closest and Most Distant Proxy Graphs Used in Dunn Index 112
A.1 MCS Distance between Node-Edge Samples (15%) 127
A.2 WGU Distance between Node-Edge Samples (15%) 128
A.3 UGU Distance between Node-Edge Samples (15%) 129
A.4 SID Distance between Node-Edge Samples (15%) 130
A.5 MCS Distance between Random Walk Samples (15%) 131
A.6 WGU Distance between Random Walk Samples (15%) 132
A.7 UGU Distance between Random Walk Samples (15%) 133
A.8 SID Distance between Random Walk Samples (15%) 134
A.9 MCS Distance between Node Edge Samples (30%) 135
A.10 WGU Distance between Node Edge Samples (30%) 136
A.11 UGU Distance between Node Edge Samples (30%) 137
A.12 SID Distance between Node Edge Samples (30%) 138
A.13 MCS Distance between Random Walk Samples 139
A.14 WGU Distance between Random Walk Samples 140
A.15 UGU Distance between Random Walk Samples 141
A.16 SID Distance between Random Walk Samples 142
A.17 MCS Distance between Node-Edge Samples (50%) 143
A.18 WGU Distance between Node-Edge Samples (50%) 144
A.19 UGU Distance between Node-Edge Samples (50%) 145
A.20 SID Distance between Node-Edge Samples (50%) 146

A.21 MCS Distance between Random Walk Samples (50%) 147
A.22 WGU Distance between Random Walk Samples (50%) 148
A.23 UGU Distance between Random Walk Samples (50%) 149
A.24 SID Distance between Random Walk Samples (50%) 150
B.1 mcs Distance between Yahoo! News Graphs 152
B.2 MCS Distance between Yahoo! News Graphs 153
B.3 SID Distance between Yahoo! News Graphs 154
B.4 mcs Distance between Node-Edge Samples (50%) 155
B.5 MCS Distance between Node-Edge Samples (50%) 156
B.6 SID Distance between Node-Edge Samples (50%) 157

Chapter 1

Introduction

In computational problems where the solution space is very large and its exploration computationally complex, there is often a tradeoff between the efficiency of a solution and its accuracy. We accept imperfect approximations every day for problems where a truly exhaustive solution is infeasible. For example, defendants are tried before a relatively small jury of their peers meant to represent the much larger community when rendering its verdict in a trial. Elected officials are selected as representatives for some set of citizens to legislate, judge, or execute the law as an entire community might if a vote were taken on every issue. Because every court proceeding cannot be tried before every member of a community and full votes cannot be taken on every issue, admittedly imperfect but efficient solutions are tolerated. The problem then becomes how to create these more efficient proxies in such a way as to minimize the deviation from an optimal solution.

This work develops the idea of Sampling for Large Input Proxies (SLIP) for different computational tasks. Ideally such a proxy retains the structural identity of the original input and can be used in its place for many problems. This identity can be seen in the enumeration of Problem Specific Features (PSF). In the example of elected officials as proxies for an electorate, the PSF's are the way in which a community would decide different issues, and the problem can be seen as selecting someone who maximizes the number of such issues on which he or

she would decide the same way. Different computational problems will have different features, of course, but the goal of SLIP should be to minimize the relative deviation from the original input's structure and therefore maximize agreement on problem specific features.

1.1 Sampling and Problem Specific Features

So how do we get such proxy for a large input? Ideally a completely unbiased subset of the original input is selected. In the jury and elected officials examples, though, we recognize that many systematic biases exist such as the requirement that members of the community be registered to participate in either process. The advantages and disadvantages of such a selection scheme are debatable, but we may certainly hope to find a more unbiased approach to find proxies for traditional computational inputs. SLIP relies on random sampling to produce proxies to large computational inputs.

1.1.1 Sampling

Sampling is used in countless disciplines to estimate various attributes of a population by measuring those same attributes for a small subset of elements from the population, and estimates obtained from the samples reliably approximate the true attributes when sound sampling methods are used. Intuitively, the accuracy improves as the number of samples approaches the size of the population, but perhaps more importantly a surprisingly small number of samples are required for an accurate approximation. The traditional sampling process can be broken down into five steps [1].

1. Identify the population

2. Identify the attributes to estimate
3. Identify the sampling technique
4. Extract estimates
5. Review the sampling process

These methods are often used to estimate the features of a population where the enumeration of the exact values for the population is either impossible or prohibitively costly. With random sampling, individuals are selected without prejudice from the population and therefore avoid systematic errors. In computation, such random sampling has been used in many application areas such as signal processing, network analysis, and machine learning to reduce the number of individuals to a much more manageable (and feasibly computable) size. Rather than selecting a subset of individuals from a population for attribute estimation, though, SLIP focuses on reducing a single, very large individual to a more compact yet equally expressive proxy. For example, a sortable permutation is reduced to a structurally similar yet much smaller permutation for sorting. Very large graphs can be reduced to structurally similar yet much more compact proxy graphs.

To accomplish this, SLIP relies on random sampling to create large input proxies. The sampling process itself differs slightly yet fundamentally from the traditional process in its identification of population and treatment of attributes for extraction and estimation. While frequently the end goal of sampling is to enumerate some features, SLIP uses random sampling to create a smaller representation whose problem specific features remain intact. Enumeration of these features is important for showing the sampling works as intended, but in the end SLIP is not used to estimate features any more than jury selection is intended to figure out the average income or age of registered voters. Rather, sampling

for large input proxies produces a representative input to be used in place of the much larger original input. For SLIP, the population is an input such as a sortable permutation or a directed labelled graph, and estimates of problem specific features are only enumerated when verification of structure retention is required.

1.1.2 Problem Specific Features

The goal of sampling a single large instance is to retain the structural identity and therefore its PSF's. Such features are defined as those properties which serve to uniquely distinguish an instance from other inputs to a specific problem. For example, the electorate for a rural community in the United States would almost certainly decide at least some issues differently than the citizens in an urban community, and we might expect their representatives to act accordingly. Similarly, computational problems have problem specific features which serve to identify the input. Sortable permutations have features of sortedness that provide some notion of the degree to which an input is already sorted, for instance. The literature is rich with examples of such properties such as the number of inversions or runs in a permutation that indicates the number of pairs of elements that are out of order. Such features are discussed in depth in chapter 3.

Problem specific features exist for other types of inputs as well. We will see that the similarity between two graphs plays a large part in various data mining tasks which require a distance measure between inputs, and we will address the question of whether the proxies of two large graphs are relatively the same distance from each other as the original inputs. Given a random sampling technique and a set of problem specific features, we will show that SLIP produces compact proxy inputs for use in several applications.

1.2 Applications

Very large computational inputs which require substantial or even prohibitive space and processing provide opportunities for applications of these ideas. Two specific areas are explored here: a pair of sorting-related problems as well as the mining of web content represented by graphs.

1.2.1 Sorting Algorithm Selection

Sorting is a thoroughly studied problem widely considered to be solved in that speed-up generally only comes with improved hardware rather than improved algorithms. Several aspects of the problem, though, remain computationally interesting if only because of their potential impact on other more computationally complex problems. Specifically, different sorting algorithms perform differently depending on how sorted the input already is. No matter how efficient a particular solution, it is not always the most efficient for every instance of input. This is an example of the algorithm selection problem: given a particular input, choose the best among several alternative solutions to solve the problem. We would like to find the best algorithm for a problem as efficiently as possible, and SLIP allows us to do that by creating proxies whose problem specific features are approximately the same as the original inputs. For any application of SLIP to a problem, though, we must identify problem specific features and a sampling method for that problem's inputs.

For sorting, many researchers have enumerated long lists of problem specific features that indicate how well a particular algorithm may perform on a given instance, but more often than not such features are more costly to enumerate than the act of sorting itself. SLIP creates input substitutes to be used to select the best algorithm without the costly enumeration of PSF's for the original input through a sampling method that preserves relative ordering in the proxies.

Experiments are conducted to compare the performance of several sorting algorithms on permutations in various stages of presortedness and to enumerate certain features of sortedness such as the number of inversions. SLIP techniques are then used to create proxies of various sizes. With these samples, the same set of features are enumerated, and the same algorithms are used for sorting. The ability of the samples to estimate the features of the much larger inputs is compared, and the relative performance of the sorting algorithms on the proxies is compared with their performance on the original inputs.

In addition, a second similar sorting problem is analyzed. The Shell sort is a multi-pass sorting algorithm employing an increment set which determines the intervals over which an insertion sort is performed for an input. For each increment in the increment set, every i^{th} element of the input is sorted as if those elements were members of their own permutation, and these increments become smaller and smaller until 1 is reached so that every element in the input is sorted together. Similar to the algorithm selection problem, different increment sets have been shown to perform differently for inputs in various stages of presortedness. SLIP is used to compare the performance of these increment sets for large inputs and their proxies.

1.2.2 Web Content Mining

In web content mining, tasks including classification and clustering are performed on web-accessible content (such as plain text, HTML, XML, and multimedia) to label and group previously unseen inputs. In classification, a particular input is assigned a label or labels often based on the category of similar inputs whose categories are already known. In clustering, several inputs are grouped into some number of categories based on some notion of their relative similarity to each other. Note that both of these tasks depend on some notion of similarity.

In fact, for both of these tasks, this similarity (or distance) between two inputs is exactly the problem specific feature we hope to retain when sampling for proxies. We know the PSF's, but what types of inputs are we sampling?

Graphs offer a particularly rich representational power that capture the inherent structure between pages on the web as well as the content within the pages themselves, but graph processing can be computationally expensive because of this same expressiveness. Other representational structures such as feature vectors whose similarity is more quickly computed are traditionally used for data mining tasks that require the PSF of relative input distance. Graphs have been shown to be at least as accurate (and often much better) than these representations in tasks such as classification and clustering [2], though, and this can be attributed to a graph's ability to capture more information about an input. However, the performance of processing such graphs severely limits its use as an efficient representational structure. Using SLIP, the input graphs created from this content can be reduced to much smaller yet still structurally similar proxies for use as substitute inputs both to classification and to clustering problems with vastly improved performance.

Two data sets of web content of different scale are used to measure the effectiveness of SLIP to create accurate large input proxies. The first includes directed labelled graphs created from the text of twenty newsgroups, and the second includes data gathered from the Yahoo! News service across seven broad categories with 41 subcategories of news articles. While the first data set represents extremely large graphs (tens of thousands of nodes with many times more edges), the second data set represents a significantly smaller (many hundreds of nodes) amount of information. The difference in the scale in the data sets reflects the ability of SLIP to produce proxies for a variety of inputs for different problems.

As with all applications of SLIP, the how and for what questions must be answered. Two graph sampling methods address the how, and for what is answered by several graph similarity metrics. Random walk sampling as well as a new hybrid node-edge random selection technique are used to create proxies. To measure their relative effectiveness, five different notions of graph distance based on the maximum common subgraph, minimum common supergraph, and a new approximation technique called the sample intersection distance are used. Results show that SLIP produces very accurate results with significantly reduced processing time and space requirements for classification and clustering.

1.3 Organization

Related research in sampling in computation as well as the use of features is surveyed in Chapter 2. In chapters 3 through 5, specific applications of SLIP are presented. First, two sorting-related problems are studied that build on the use of SLIP to approximate features of sortedness. Large permutations along with their sampled proxies are used in experiments with several common sorting algorithms as well as multiple Shell sort increment sets to show that indeed appropriately designed sampling not only retains relevant problem specific features but also be can used to select the most appropriate sorting algorithm or Shell sort increment set for a given instance more efficiently.

Two data mining applications are then explored in chapters 4 and 5. In chapter 4, a collection of twenty newsgroups is converted to a collection of graphs based on word occurrence and proximity. These graphs and their samples are then classified using an adapted k-nearest neighbors method. Such a method determines an input's label by essentially taking a vote among it's k nearest neighbors according to some measure of similarity. A variety of graph distance

metrics and sampling schemes (including some novel techniques) are used, and classification accuracy of the proxies is compared with that of the original inputs.

In chapter 5, graphs are created from a collection of Yahoo! News articles covering seven general categories and 41 subcategories. The original graphs as well as the proxies created through SLIP are then clustered using a global k-means algorithm tailored for graphs using a variety of distance metrics and sampling methods. The retention of problem specific features in the proxies of these smaller inputs is examined, and clustering performance of the proxies compared to the original inputs is analyzed. Results from the classification and clustering of both data sets show that SLIP is an effective means of improving the efficiency of web content mining without losing the expressiveness of the graph data structure. Finally, concluding thoughts and directions for future work are discussed in chapter 6.

Chapter 2

Survey of Related Literature

There is a great deal of related work in applications of sampling in computation as well as the use of input features in computation. Because of the vast related literature in sampling in computing, we will examine only some of the more relevant examples in application areas such as signal processing, networks, data and information management, and machine learning. Features are also another very general area with a great deal of related work, and again we will restrict our focus to the more relevant examples.

2.1 Sampling in Computation

Sampling is an extremely common approach to estimating an attribute of a population through calculating the same attribute for a subset of the population. Numerous random and non-random methods exist, but the aim of each is to produce the most accurate approximation possible. Sampling is a common technique in computation because when used correctly it provides very accurate results with large improvements in efficiency, and it is used whenever it is difficult or impossible to process all members of a population. In addition, an exact answer is not always required, and an estimate of a given feature based on sample will suffice.

The purpose of sampling can also be viewed in two different ways. In the first, an actual attribute is estimated for the population. In the second, the goal of the

sampling is to create a proxy for the population. SLIP falls into this latter category. In order to prove its effectiveness, though, the problem specific features of the proxies created through SLIP must be tested against the same properties for the originals. Therefore we will survey several different uses of sampling across several subfields of computer science including signal processing, networks, data management, and machine learning.

2.1.1 Signal Processing and Analysis

In signal processing, inputs such as video and audio signals are analyzed, interpreted, and manipulated. Sampling is frequently used in this domain to reduce a continuous input into discrete units, and we see applications daily in audio, speech, and video processing. Nyquist and Shannon introduce the foundation of signal sampling [3, 4]. Essentially continuous signals of a given bandwidth may be perfectly reconstructed if the sampling frequency is at least twice the signal bandwidth. Effects such as aliasing are caused by undersampling or poor reconstruction of the image from the sample. Signal sampling attempts to convert one type of signal (continuous) into a discrete structure. There are several texts and journals providing in-depth coverage of signal processing and analysis including [5, 6].

2.1.2 Networks

Network data comes in many forms. Each layer of the common network stack has an associated information unit [7]. For example, at the network level, it may be the packet while at the application layer it may be discrete chunks of context or application-specific data. In addition, the network topology itself can be seen as the data. The latter example of network data has become increasingly important as the size of networks continues to grow.

Sampling techniques are actually used at multiple network layers to handle the overwhelming volume of data and sizes of networks. Patterns in packet traffic are often analyzed for various types of malicious activity such as denial of service attacks, intrusion detection, and network flow distributions [8, 9, 10, 11]. In such cases, randomly chosen packets are chosen for analysis because the population of packets is simply too large. At the application layer, sampling is used to select a subset of individual emails or web pages on which to conduct analysis for things such as SPAM detection or other information mining tasks. This particular area can also be classified as web content, and related work in this domain is discussed later in chapters 4 and 5.

The layout and evolution of network topologies have also been analyzed using sampling techniques. With very large, dynamic networks, there is a natural mapping of ideas with graph data structures, and this has been exploited. In [12], the use of random walks in peer-to-peer (P2P) networks are explored for routing and mapping. Cooper et al. use random sampling of regular graphs to model networks [13]. Krishnamurthy and others explore sampling techniques to reduce the size of network topologies including the Internet for faster simulations [14, 15]. In network routing, the performance of distributed hash tables benefits from random sampling [16]. Other work has focused on using sampling to visualize large networks [17]. The use of sampling in more general network problems such as cut, flow, and design issues are explored in [18]. Much of this work and other uses of sampling in processing and analyzing networks is closely related to the application of SLIP to web content mining.

2.1.3 Data and Information Management

In data and information management, we see a wide array of sampling applications as the size of data pushes the limits of reasonable computation times and

storage limits. For web-accessible data, Bar-Yossef uses sampling to estimate very accurately the coverage of a search engine's index without having to search exhaustively [19]. In [20], sampling is used for aggregate queries on large data sets using random walks. For streaming data, sampling has been used effectively and efficiently to monitor properties of the stream through sampling [21].

More traditional relational database tasks have also benefitted from sampling techniques. Chaudhuri et al. explore the difficulties of using sampling with join trees and develop a more efficient way to use sampling to speed up such operations while Naughton and Seshadri use estimates of the size of various relational operators such as projection to improve performance [22, 23]. Other uses of random sampling in relational databases are explored in [24, 25]. Sampling has also been used in spatial databases to improve performance of retrieval [26]. Seshadri et al. explore sampling issues in parallel database systems [27].

Several data mining tasks take advantage of the ability of sampling to approximate a search space. Brin and Page use sampling to explore otherwise prohibitively large search spaces of rule sets [28]. Schutze and Silverstein explore projections for more efficient document clustering [29]. Other applications in data mining can also be considered machine learning, and they are covered in section 2.1.4.

2.1.4 Machine Learning and Evolutionary Computation

Machine learning and evolutionary computation are other areas where sampling is used. Many of the inputs to problems in these areas contain a large number of individuals on which to train and test processes, and sampling provides a way not only to reduce training and testing times but also a way to turn a limited number of examples into training and testing sets. Bootstrapping processes also

rely on sampling to initially seed groups with individuals. In fact, many evolutionary computing tasks begin with randomly chosen individuals from which to evolve a more optimal solution.

In addition to reducing the number of individuals to a more manageable size, various sampling methods are also used to reduce the number of features of an individual. Here features can mean something different than the problem specific features of SLIP. For example, in some machine learning tasks, an individual may have many characteristics (features) all of which may or may not have some bearing on the learning task. Sampling can be used to select a subset or many subsets of these features in order to find out which such subset is optimal.

2.1.5 Graphs

There has been a variety of work in graph sampling in just as many application areas. While some remain at least in part theoretic [18, 30], other work using graphs based on real data has been done [30, 31]. Bar-Yossef et al. accurately estimate the true coverage of a search engine's index through sampling techniques [19], and Stutzbach et al. use sampling on dynamic graph data which changes over time. As mentioned previously, networks lend themselves well to the graph structure, and some applications of graph sampling appear in peer-to-peer network research [13]. Leskovec et al. have shown a variety of sampling techniques do indeed preserve several graph theoretic features such as in-degree distribution, out-degree distribution, sizes of weakly and strongly connected components, hop-plots, and others [30]. Random walks are a common method to explore (and sample) large graphs, and there have been a variety of applications as well as explorations of optimal parameters to these walks [32, 30]

2.2 Input Features

As described in the introduction, features of an input can differentiate one instance from another. Like fingerprints, they are what can make a particular input unique for some problem. In computation, the term feature can mean many things. In this work, a feature refers to a problem specific property of an input that differentiates it from other inputs with respect to some problem. We survey various notions of features for the types of inputs for the problems we consider in applications of SLIP. It is important to note that the independence of some input feature from a problem does not imply that the feature does not have an impact on the performance of solutions to that problem.

Feature extraction commonly involves machine learning techniques for document, image, or other complex resource processing. For documents, words are analyzed to construct feature vectors that provide a means to compare documents for similarities. For more complex document processing, Brin introduces the idea of sampling to explore the solution space for rule sets in order to avoid an otherwise prohibitively costly search [28]. Feature extraction in image processing is similar in that it is more specific than a general framework for inputs to a given computational problem. Other work has focused on approximating features in a specific domain as well. Srivastava and others approximate the properties of queries (in particular, the maximum distance between inversions) by using windows in a data stream and extrapolation using a probability distribution [21]. Here we consider features related to the problems discussed in the later chapters.

2.2.1 Features of Sortable Inputs

The types of computational inputs are endless. They can include but are certainly not limited to a single number, a string of characters, a long permuta-

tion of some set of numbers or strings, trees, graphs, and large corpora of text. Each such structure has its own features independent of any problem definition. For example, a number can have a magnitude, a string can have a length, and graphs have densities and edge probabilities. For the sorting problem, the inputs are sets of orderable keys where the keys may represent numbers, strings, or anything else with some notion of order attached to it. There has been a great deal of work in features of sortedness [33, 34, 35, 36, 37, 38]. This work helps to define notions that characterize how sorted a permutation already is. Such measures as the number of inversions, number of sorted subsequences, and the maximum distance between an element and its sorted position are defined in detail in chapter 3.

2.2.2 Graph Features

There are virtually countless features of graphs, especially when application or domain specific features are considered. While this work will focus on the problem specific features that help to distinguish one graph from another in the form of distance or similarity measures, other features of graphs range from the relatively simple to the much more complex. For example, graph density and edge probability help to identify a graph, and they can play a role in the performance of algorithms for problems such as Minimum Spanning Tree. Other features such as the number of weakly connected components, strongly connected components, and cliques of different sizes become increasingly more difficult to compute, but these features can also serve to identify an input as unique or at least as belonging to a type of graph.

The graph classification and clustering problems depend on a particular PSF that provides some notion of graph similarity or distance. That is, given two graphs, how similar (or different) are they? Related work in string and tree

matching [39, 40] has been extended to graphs and other types of inputs such as point sets [41, 42]. Several approaches to graph distance have been developed, and many of these are discussed in more detail in Chapter 4. Hierarchical discrete relaxation and graph matching are discussed in [43, 44] while [45] provides an overview of many graph similarity techniques. Graph intersections and maximum common subgraphs are discussed in [46, 47] while graph unions and the minimum common supergraph are covered in work such as [48, 49, 50]. General approaches to distinguishing features of graphs for data mining are also discussed in [51, 30].

2.3 Relationship with this work

Much of this work is related to SLIP and its applications through either its use of sampling or its use of similar inputs, but little previous work is directly related to both notions that are central to SLIP. Sampling for Large Input Proxies represents a new approach to improving solutions to problems such as algorithm selection and data mining. Related work to these specific problems are discussed in Chapters 3 , 4, and 5. We will see that SLIP does indeed allow for improved computational efficiency through sampling for large input proxies.

Chapter 3

Sorting Algorithm Selection

3.1 Introduction

For a given problem, there often exist several alternative solutions which perform differently depending on characteristics of the input. For example, various sorting algorithms to order some set of permuted keys will perform differently depending on how ordered such an input already is. Such problem specific features (PSF) of computational inputs can uniquely identify an input with respect to a problem, and these features are often what determine which particular algorithm from many alternatives will most efficiently solve the problem for a given input. In such cases, if the relationship between PSF's and efficiency of various solutions is known, the best algorithm can be selected for that input.

This chapter proposes an efficient solution for two sorting-related problems using sampling for large input proxies (SLIP) – sorting algorithm selection and Shell sort increment set selection. We will define several PSF's for these sorting problems, and we show that not only do proxies of large inputs retain the features of the original inputs but that these proxies can be used to find more efficiently the best algorithm or increment set for a given input. Methods to select such proxies for and to estimate problem specific features of the sorting problem are introduced. PSF's for both the original inputs and their proxies are analyzed, and SLIP proxies are used as more compact but similarly expressive substitute inputs to both sorting problems.

3.1.1 The Sorting Problem

Sorting is one of the most thoroughly researched problems in computer science. Given an input of N items each with key K_i, sorting involves finding the permutation $p(1), p(2), p(3), ..., p(N)$ of this collection such that the keys of all elements are in non-decreasing order $(K_{p(1)} \leq K_{p(2)} \leq ... \leq K_{p(N)})$ [38]. Over the years, researchers have proposed a large number of algorithms and their variations to solve this problem in the most time and space efficient manner. Knuth discusses over twenty of them in his *The Art of Computer Programming* [38], and most of these algorithms fall into one or more of the following classes: insertion sort, exchange sort, selection sort, enumeration sort, or a special purpose sort. Every sorting algorithm has its advantages and disadvantages with respect to ease of implementation as well as time and space efficiency. Some well known comparison-based efficient algorithms include quick sort, heap sort, Shell sort, and merge sort. The performance of these algorithms for a particular input often depends on the PSF's known as features of sortedness.

3.1.2 Problem Specific Features for Sorting and SLIP

If inputs may be identified in terms of their problem specific features, then these features are useful in several applications such as the algorithm selection problem. For sorting, a permutation of orderable keys has several features of sortedness defined in the literature [38, 36] where sortedness refers to how ordered a permutation already is. For example, the number of inversions in a permutation is equal to the number of pairs of elements at i and j in a permutation where the i^{th} element is ordered after the j^{th} element but $i < j$. In the sequence (2,5,4,3), there are 3 inversions - (5,4),(5,3), and (4,3).

Problem specific features can help close the gap between problem complexity and algorithm efficiency, but enumerating the features is often at least as costly

as solving the problem itself. For example, calculating the number of inversions in a permutation can be calculated in $N \lg N$ time where N is the number of orderable keys of the input while the sorting problem itself is also $N \lg N$ [1]. In these cases, an accurate estimate of such features can be useful. In addition, an exact enumeration of input properties is frequently unnecessary. That is, an approximation of the problem specific features provides accurate enough information to make a decision.

Part of the focus of this chapter is the enumeration of PSF's for proxies of large inputs to compare their relative accuracy to the PSF's of the original inputs. Given an input of some orderable keys, a model for restricted sampling is outlined such that input proxies retain the original input's features of sortedness. It is critical of course that any such sampling method used in any SLIP process maintain the structure of the original input so as to preserve the problem specific features as well. For example, when sampling a permutation to be sorted, the relative order of the keys should be maintained in the sample. That is, if some element i comes before element j in the original input, that ordering should be maintained in any proxy created through SLIP. In other words, if the sampling method simply picks 10% of the keys in random order for insertion into the proxy, the proxy may likely bear little resemblance to the original input with respect to the features of sortedness. The sampling process is explained in more detail in section 3.2.2.

3.1.3 Sorting Algorithm Selection and Shell sort Increment Sets

The second focus of this chapter is the use of SLIP to select the best algorithm for sorting as well as the best increment set among many for the Shell sort. The sorting problem has been shown to be of $N \lg N$ complexity (for comparison-based

[1] $\lg$ refers to log_2 throughout this work while $\log$ refers to log_{10}.

sorting), but actual algorithm performance often deviates from this in the best and worst case due to various features of the input. For example, quick sort has an average performance of $N \lg N$, but the run time can degrade to N^2 given particular permutations with certain characteristics. Insertion sort can run in linear time for an already sorted input, but its runtime also degrades to N^2 when an input is in reverse order. Often times an algorithm *A* which outperforms some other algorithm *B* on a certain kind of data can be outperformed by algorithm *B* on different data. The notion of having multiple algorithms that perform differently depending on the problem input leads to what is known as the algorithm selection problem [52].

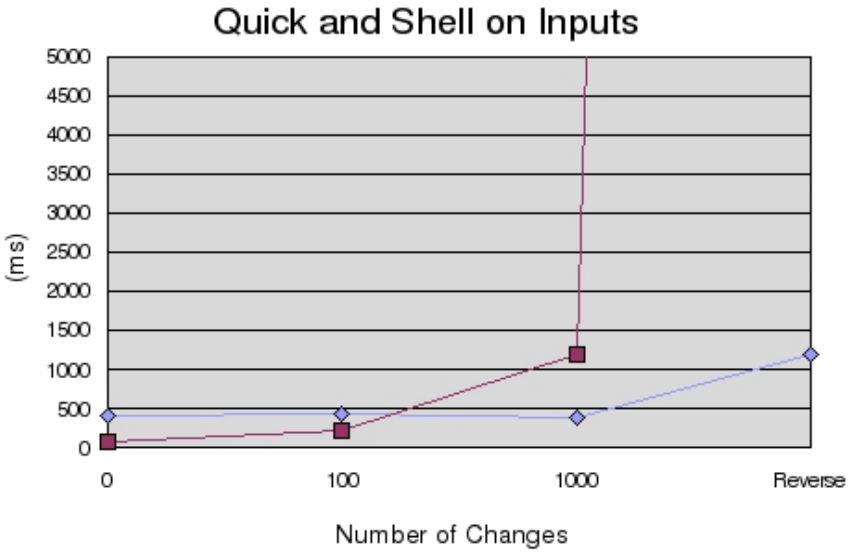


Figure 3.1: A comparison of Quick and Shell sorts on different types of inputs. The performance of Shell sort severely degrades when more than 1,000 random element exchanges are performed.

For many such problems, it is common to simply select an algorithm which *generally* solves the problem most efficiently. For sorting, quick sort often performs best in the average case so it is used for all cases. Of course, we would ideally like to select *specifically* the most appropriate algorithm for each instance of the problem. For example, in Figure 3.1, we see that one version of a Shell

sort outperforms a particular implementation of the quick sort in some cases, but it is drastically worse in other cases (such as when the input is in reverse order). Ideally we would also like to select the best solution without adding prohibitive amounts of computation to the process. We will explore the use of SLIP and PSF's for the sorting algorithm selection problem through experiments featuring several sorting algorithms including quick sort, merge sort, heap sort, and Shell sort. PSF's are compared, and the samples are used to determine the best algorithm for sorting the original input.

A related sorting problem involves the Shell sort. This particular sorting algorithm relies on a set of increments to perform a multi-pass insertion sort, and its performance varies widely depending on the values of the increments. Several well-known increment sets including Knuth, Sedgewick, Gonnett and Shell (defined in Section 3.4) are used to sort both inputs and their proxies created through SLIP. The PSF's are compared, and SLIP's ability to create proxies which predict the best increment set is analyzed.

3.1.4 Organization

The rest of the chapter is organized as follows. Section 3.2 presents motivation and experimental results for sortedness feature approximation. Section 3.3 describes an application where input proxies are used to select the best sorting algorithm for a given input, and section 3.4 describes similar results comparing the performance of different increment sets for the Shell sort using sampling. Conclusions are discussed in section 3.5.

3.2 Approximating Features of Sortedness using SLIP

The problem specific features we wish to preserve with SLIP for the sorting problem are known more broadly as features of sortedness. There has been a

great deal of work in identifying these features [38, 35, 36, 34, 53, 33], and much of this chapter concentrates on enumerating and approximating these PSF's of sorting inputs. A set of eleven well-researched features are discussed in [35] as an extension of some of the material presented in much of the sorting literature [38, 33]. With respect to certain features of sortedness, certain algorithms are adaptively optimal in that they may perform the fewest possible comparisons within a constant factor given a particular feature or set of features [35]. Because these problem specific features lie at the heart of why certain algorithms outperform others in certain problem instances, a user can identify the most appropriate sorting algorithm if he or she knows the features of the input. This information is rarely known, though, and each sortedness feature requires at least linear and often polynomial time to enumerate explicitly. Thus extraction of the features of sortedness can actually be more costly than the sorting process itself.

3.2.1 Features of Sortedness (or Problem Specific Features of Sorting)

Several attempts such as machine learning approaches [37] and adaptive sorting techniques [35] have attempted to sort faster by taking advantage of input features such as the number of inversions, number of sorted subsequences, and other input characteristics. Results often provide a way to assess an algorithm's optimality with respect to certain measures of sortedness, but these require foreknowledge of each instance or calculation of such properties on the fly. Other research has focused on refining certain algorithms such as the Shell sort. It is not entirely clear what causes certain increment sets to perform differently for different types of inputs for this algorithm, but Sedgewick and others continue to search for optimal sets of increments and to provide tighter bounds on the efficiency of the sorting algorithm [53]. If we can approximate these features

more efficiently, though, we can show that the proxies can be used in the sorting algorithm selection problem.

We will see that SLIP indeed produces proxies of sorting inputs that retain these features of sortedness, and these proxies can be used to identify the PSF's of the input in a much more efficient manner. To show this, we sample large permutations ranging from sorted to nearly sorted to random that contain a range of features of sortedness such as number of inversions, the greatest distance between two inversions, the maximum distance that an element must travel to its position in the sorted permutation, and the number of runs (defined below). These particular features are considered representative of the different notions of how sorted an input already is, and they are chosen here for that reason. Many other measures exist, but they mainly consist of incremental adaptations of these more fundamental notions of sortedness.

The number of inversions is defined as the total number of pairs of elements which are out of order. For an input of size n, the maximum number of inversions is

$$\sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} \tag{3.1}$$

The largest distance between any pair of inverted elements is defined as the maximum inversion distance, and it is bounded by the size of the input. The maximum distance an element must travel refers to the distance an element must be shifted before it is in its ordered place. This differs from maximum inversion distance because the pairs considered are not all inverted element pairs but rather the pair of starting and final element positions. This measure is also bounded by the size of the input. Finally, the number of runs in a permutation is the number of ordered subsequences contained in the permutation. In the worst case, a reverse-ordered permutation of size N has N runs while an already

ordered permutation has 1 run. Using these PSF's, sorting input proxies are analyzed for the same features as the original input relative to size. Table 3.1 contains data for four different types of permutations and thirty samples taken from each, and it is discussed in detail in section 3.2.4.

3.2.2 Sampling Procedures

Inputs are defined as a set of orderable keys, and our samples will consist of an order-preserved subset of these keys. These keys can represent anything such as numbers, strings, or more complicated objects, of course, but these experiments use permutations of integers. Here "order-preserved" means that if some element i comes before some element j in the original input, such an ordering is also guaranteed in any proxy created with SLIP. We define the attributes to be approximated as the problem specific features defined in the previous section. This is, of course, a relatively simple scheme, but the sampling technique itself contains the most crucial part for feature retention and approximation. In-order random sampling without replacement is used in order to avoid introducing new relationships between elements that do not exist between them in the population. Because we are measuring PSF's, it is critical to maintain the structure of the specific input instance in samples without introducing new information or relationships.

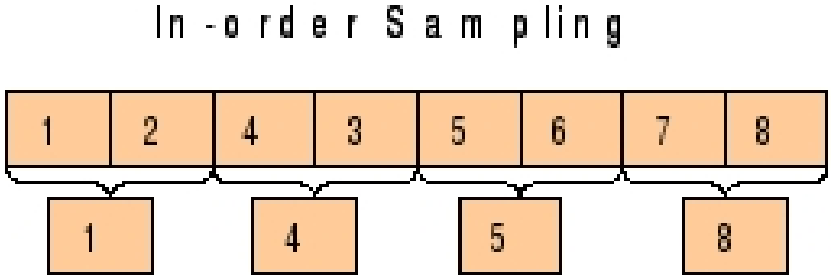


Figure 3.2: Example of in-order sampling method for a permutation.

The random sampling is "restricted" in the sense that the relative ordering of the input is not disrupted. For sorting, this order is fundamental to most notions of sortedness. Similarly for any applications of SLIP, the random sampling should not introduce new features to the problem. For example, for a 50% sample, we simply take a random element from every two elements. In figure 3.2, we can see how the four elements $(1, 4, 5, 8)$ are sampled from the input $(1, 2, 4, 3, 5, 6, 7, 8)$ while their relative order is maintained. If an element k_i comes before element k_j in the input, the elements retain the same relative order in the sample. This necessary restriction produces a proxy that resembles the input itself with respect to features of sortedness. Equations 3.2, 3.3, and 3.4 show the number of possible proxies with and without replacement as well as the possible number of proxies under the SLIP scheme. For all, r is the sample rate while n represents the number of elements in the input. The number of potential proxies for sampling with replacement is shown in equation 3.2, the number of potential proxies for sampling without replacement is shown in equation 3.3, and the number of proxies possible under the order-preserving sampling scheme of SLIP is shown in equation 3.4

$$n^{r \cdot n} \tag{3.2}$$

$$\binom{n}{r \cdot n} \tag{3.3}$$

$$\left(\frac{n}{r \cdot n}\right)^{r \cdot n} \tag{3.4}$$

For the sorting problem, it is crucial that any sampling method maintain the relative order of these elements to preserve problem specific features such as number of inversions and runs. Thus only a fraction of the possible permutations of subsets of a required size are available for selection as proxy. Selecting an r

and thus a sample size is only approached experimentally here. A theoretical analysis of optimal sample sizes for different types of inputs of different sizes and characteristics is beyond the scope of these chapters, but for each type of data several sample rates are used to show at least experimentally the tradeoff between efficiency and accuracy.

3.2.3 Experiments

For the feature approximation experiments, each input begins as a perfectly sorted list of numbers between 1 and 100,000. We will use larger inputs in the algorithm selection experiments, but because some of these features are very costly to calculate, we use smaller permutations here. Once these sorted lists are created, we perturb them in two ways. First a predetermined number of runs is created by moving subsequences around so that 1, 2, 4, and 8 runs result. From these four input types, some number of exchanges (between 10 and 10,000) of random elements are performed. That is, for each possible number of runs, 10^i random flips are made where i ranges from 1 to 4. By varying the number of runs and the number of random flips, a variety of inputs ranging over the notions of sortedness are created. For these feature approximation experiments, the SLIP process uses sample rates of 1, 5, and 10% in order to get an accurate picture of the tradeoff between accuracy in retention and the efficiency in sampling and extraction.

3.2.4 Results

Table 3.1 presents a summary of the empirical data for inputs with one run and a variety of random flips. Details of a specific example are provided, but first the general results are described. The numbers represent averages of 30 samples drawn from different inputs. The data show that the proxies accurately retain

the measures of sortedness from the original inputs. As expected, as the sample size grows, the feature approximation becomes even more accurate. Likewise, as the permutation goes from nearly sorted to 10,000 random perturbations, the features become more and more similar between the input and samples. These reductions in variation for both larger samples and more randomized input are intuitively rooted in the idea that deviations from sortedness are more likely to show up when either more of the input is sampled or there are more perturbations from which to pick.

In the tests for a nearly sorted collection of 100,000 elements (with only 10 randomly flipped elements), the features measured are more skewed because it is likely few if any of the randomly flipped elements show up in the sample. This would of course make the sample completely sorted. There is more deviation in the measures for the smaller samples, but this is expected. Rates relative to the size of the input are also listed in the table because these sortedness features are measured with respect to the size of the input (or sample). Overall, for each permutation and set of sortedness features, the features of the samples become a more and more accurate reflection of the input's features as the sample size grows.

Because each feature is measured with respect to the size of the input (or sample size in the case of the samples), a relative measurement of the estimates provides a better picture of how well various sample sizes perform for different features. For the last three features of sortedness we measure, the rates in the figure present a more accurate indicator of the similarity between the samples and the input because these measures are relative to the size of the list sorted (100,000 for the input but different sizes for the various samples). For the number of inversions, the rate is also a meaningful measure. Because the number of inversions can actually be quadratic of the input size, though,

Changes	Size	# Inversion	Max Dist	Max Trvl Final	# Runs	Inv Rate	Dist Rate	Trvl Rate	Run Rate
10	**100000**	**762002**	**94192**	**87858**	**20**	**7.62**	**0.94**	**0.88**	**0**
	1000	86.533	79.867	79.867	0.267	0.087	0.080	0.080	0.000
	(std dev)	*176.816*	*162.982*	*162.982*	*0.521*	*0.177*	*0.163*	*0.163*	*0.001*
	5000	1691.633	1559.233	1559.233	0.867	0.338	0.312	0.312	0.000
	(std dev)	*1625.978*	*1554.877*	*1554.877*	*0.819*	*0.325*	*0.311*	*0.311*	*0.000*
	10000	8234.567	4551.167	4467.833	2.233	0.823	0.455	0.447	0.000
	(std dev)	*6931.570*	*2772.530*	*2707.316*	*1.524*	*0.693*	*0.277*	*0.271*	*0.000*
100	**100000**	**6132214**	**97762**	**84560**	**199**	**61.322**	**0.978**	**0.846**	**0.002**
	1000	626.767	364.733	357.600	1.867	0.627	0.365	0.358	0.002
	(std dev)	*584.912*	*269.665*	*264.275*	*1.432*	*0.585*	*0.270*	*0.264*	*0.001*
	5000	13724.633	3654.233	3263.700	8.800	2.745	0.731	0.653	0.002
	(std dev)	*5173.969*	*771.579*	*719.648*	*3.178*	*1.035*	*0.154*	*0.144*	*0.001*
	10000	63597.600	8369.833	7641.767	19.933	6.360	0.837	0.764	0.002
	(std dev)	*14333.335*	*905.874*	*843.215*	*3.750*	*1.433*	*0.091*	*0.084*	*0.000*
1000	**100000**	**66726712**	**99644**	**95658**	**1963**	**667.267**	**0.996**	**0.957**	**0.020**
	1000	7333.600	891.233	823.400	19.767	7.334	0.891	0.823	0.020
	(std dev)	*2312.302*	*76.130*	*91.200*	*4.408*	*2.312*	*0.076*	*0.091*	*0.004*
	5000	169803.367	4877.533	4471.400	100.067	33.961	0.976	0.894	0.020
	(std dev)	*17490.349*	*84.557*	*210.473*	*8.851*	*3.498*	*0.017*	*0.042*	*0.002*
	10000	666638.867	9857.467	9216.567	196.467	66.664	0.986	0.922	0.020
	(std dev)	*45244.563*	*65.241*	*304.339*	*13.093*	*4.524*	*0.007*	*0.030*	*0.001*
10000	**100000**	**572251590**	**99987**	**99727**	**16442**	**5722.516**	**1.000**	**0.997**	**0.164**
	1000	58784.733	986.467	935.600	168.233	58.785	0.986	0.936	0.168
	(std dev)	*4013.646*	*8.966*	*28.686*	*9.744*	*4.014*	*0.009*	*0.029*	*0.010*
	5000	1431734.867	4983.533	4816.333	818.800	286.347	0.997	0.963	0.164
	(std dev)	*48904.094*	*11.907*	*84.899*	*22.813*	*9.781*	*0.002*	*0.017*	*0.005*
	10000	5804830.200	9987.633	9809.267	1663.367	580.483	0.999	0.981	0.166
	(std dev)	*133782.807*	*8.401*	*130.002*	*28.026*	*13.378*	*0.001*	*0.013*	*0.003*

Table 3.1: Comparison of Sortedness Features for Inputs and Samples. Each block represents an input with 10^i random perturbations as well as the averages of 30 proxies of size 1, 5, and 10%. For the inputs and proxies, the features of sortedness are compared both in the absolute and in the relative rates. Because all but the number of inversions is bounded by the size of the input (or proxy), the rates provide standardized measures. For inversions, though, the figures must still be scaled by a factor of the sample rate for a more standardized comparison. We see that as expected larger proxies provide more accurate approximations, but even very small proxies with respect to the size of the input provide accurate notions of the features of sortedness

the rates should be scaled to the same size in order to reflect the relationship between the sample and the input accurately. Progressing through the data from nearly sorted (10 changes) to random (10,000 changes), we see how each feature is more closely approximated by all sample sizes. In addition, we can also see how in general the progressively larger samples also more closely estimate the measured properties.

An example will help make sense of the numbers, but we need to understand how to compare the number of inversions for different sizes of permutations. Because inversions are quadratic with respect to the size of the input (that is, a sequence with 10 times as many elements can have approximately 100 times as many inversions), the proxies' inversions must be multiplied by the square of the sampling factor (10^2 for a 10% sample, 100^2 for a 1% sample). Other properties such as the maximum distance between inversions are linear with respect to the size of the permutation so that no additional scaling is required.

Two types of numbers from the sample data are compared in Table 3.1: intra-permutation where numbers from different sample rates of the same permutation are compared and inter-permutation where numbers from the same sample rate but from different permutation types are compared. For the former, the number of inversions for the 1 and 10% samples for the 1,000 perturbation input provides a good example. The original input has 66,726,712 inversions. The average number of inversions for the smaller sample size is 7,333.6 while the larger sample averages 666,638.9 inversions (73,336,000 and 66,663,890 scaled to the same size as the original input). Both of these numbers provide an accurate estimate of the number of inversions expected in the original input (and the average of the larger sample proves to be very accurate). In addition, the variation with the sample set is more stable in the larger sample. If scaled to input size, the standard deviations of 2,312.3 and 45,244.6 show us that any

given sample with a 10% sample rate is more likely to be closer to the mean than a sample with a smaller sample rate.

The number of inversions for the 5% samples of the 10-flip and 1,000-flip inputs provides an illustrative example for inter-permutation comparisons. For each input, there are 762,002 and 66,726,712 inversions respectively. For the nearly sorted input, the 5% sample averages 1,691.6 (a scaled value of 676,653.2) inversions while the random permutation averages 169,803.4 inversions (scaled value of 67,921,346.8). Both 5% samples provide good estimates for the number of actual inversions in their respective original inputs, but the sample for the more random permutation is most accurate. The variance within the sample set measures the accuracy in a different way. The scaled standard deviations for the nearly sorted input (650,391.2) and for the 1,000-flip (6,996,196) show that variation from the expected case for any given sample is much larger in the 10-flip input. Both of these inter-permutation comparisons reveal a common characteristic about sortedness features and sampling: the more sorted an input is, the more deviation there will be in the estimate for that feature from original to proxy.

Both the specific numbers as well as the general trends in the sorting feature experiments reveal two facts. First, as expected, larger samples do a better job of approximating the problem specific features of the inputs, but even small samples can be very accurate in their estimates. This of course makes sense when carried to the extreme of larger and larger sample rates that eventually end with 100% sampling. Second, samples more accurately estimate features of the input when there is more disorder in the input. That is, the more perturbations from sorted, the more likely these deviations will show up in a given sample. For example, a 5% sample of a list that is perfectly sorted except for one element has only a 1 in 20 chance of detecting that single out of order element.

In these experiments we see that SLIP can be used effectively to create proxies for inputs of the sorting problem to estimate the features of sortedness. These PSF's can of course be enumerated to get an estimate of the original input's features without fully calculating them for the full input, but there is an even more powerful use. Because the features are implicit in the structure, these proxies can be used for a variety of problems related to sorting including the algorithm and Shell sort increment set selection problems.

3.3 Sorting Algorithm Selection

If features of sortedness are retained in significantly smaller samples of inputs to the sorting problems, then the samples may be used wherever these features are considered useful. One such application is the sorting algorithm selection problem. The algorithm selection problem generally involves picking the most efficient algorithm among many for a particular instance of a problem [52]. The sorting algorithm selection problem more specifically requires selecting the best sorting algorithm given a particular input.

3.3.1 Sorting Algorithms

The four sorting algorithms analyzed here are quick sort (QS), merge sort (MS), Shell sort (SS), and heap sort (HS). The efficiency of each algorithm is well-studied [38, 53], and each has its own relationship with respect to the presortedness of its input. For example, the insertion sort algorithm is optimal with respect to the number of inversions. That is, if there are no inversions, then insertion sort runs in linear time. As we have seen, however, calculating many features of sortedness such as the number of inversions takes more time than sorting. For these experiments, each algorithm is run on various inputs of size

5,000,000 as well as samples of size 5, 10, and 25% taken from these inputs. For each, the runtime measured in milliseconds is recorded and compared.

3.3.2 Performance on Original Inputs

The sorting algorithms perform as expected on the various inputs. For the already sorted and almost-sorted (only 10 random element swaps), we see that the Shell sort is the most efficient. Because it is based on the insertion sort, we would expect it to perform well in such cases. The other algorithms follow suit – quick sort is fastest for more random permutations, and merge sort and heap sort are steady across all degrees of sortedness. We notice that as the permutation becomes more random, the Shell sort starts to fall behind quick sort and eventually even merge sort in efficiency.

Rdm Flips	QS	MS	SS	HS
0	8011	8757	3696	13198
10	7653	8988	4890	13232
100	7179	9374	6107	13194
1000	8124	9696	7947	13168
10000	7622	9906	11101	13227

Table 3.2: Performance Summary for Sorting Algorithms

In Table 3.2, we see the evolution of the performance rankings as the permutations become more random. These results reflect what is known about the relative performance of these well-researched sorting algorithms. The relative rankings of the sorting algorithms for each degree of sortedness will form the basis for comparison of the proxies' performances.

In Figure 3.3.2, the performance of the four sorting algorithms is compared over a variety of number of runs and number of random perturbations. As the number of perturbations and runs grows, we see that Shell sort goes from the

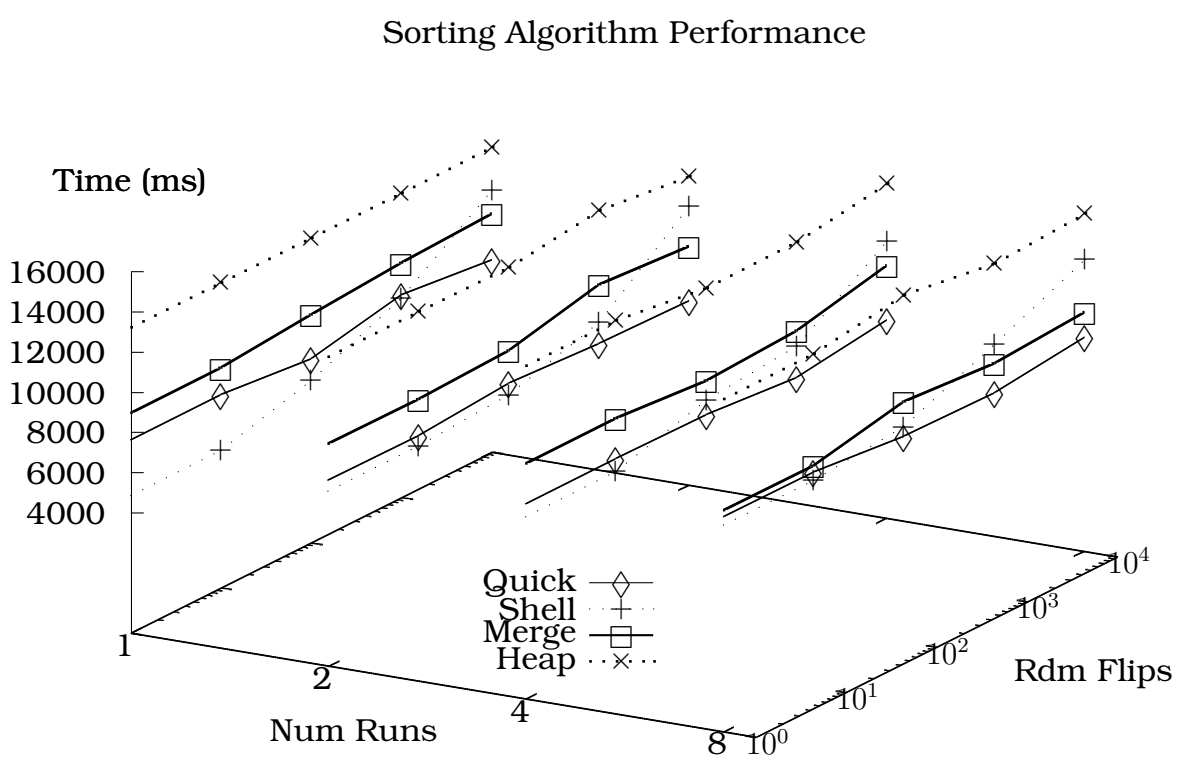


Figure 3.3: A comparison of sorting algorithm performance with 5,000,000 elements. For each type of input (1, 2, 4, or 8 runs), Shell sort degrades in performance from best to worst. The other algorithms remain fairly steady across all ranges of sortedness with some slight decrease in performance as the input becomes random.

most efficient to the least efficient while quick sort remains consistent in performance to rise to the top. Heap and merge sorts are also consistent in performance, and their rankings change little except when Shell sort begins to degrade in performance as the inputs become more random.

3.3.3 Performance on Samples

	Rdm Flips	QS	MS	SS	HS
(Orig)	0	8011	8757	3696	13198
(Sample)		287.9	339.63	143.7	515.83
	10	7653	8988	4890	13232
		289.57	340.8	142.5	520.3
	100	7179	9374	6107	13194
		288.8	348	165.57	519.6
	1000	*8124*	9696	*7947*	13168
		283.57	363.1	206.53	518.8
	10000	*7622*	9906	*11101*	13227
		288.93	378.77	283.47	513.9

Table 3.3: Original versus Sample Sorting Performance. 5,000,000 elements, one run. As the inputs become more random, we see that the quicksort and shellsort begin to swap place as the most efficient. The emphasized times show the areas where for inputs with one original run, the proxy is a bit deceptive with respect to the actual performance of the shellsort.

Running the same sorting algorithms against samples of 5, 10, and 25%, we see that the performance of an algorithm of a much smaller sample is an excellent indicator of efficiency on the original input. In Figure 3.3, the performance numbers are compared for each algorithm on the original input and on a proxy of 250,000 elements (5%). In most every case, the performance ranking on the sample predicts the ranking on the original input. In a few cases, however, we see that Shell sort and quick sort are extremely close in performance. In these cases, the sample time for Shell sort is often less at least in part because of the performance improvements gained for smaller inputs.

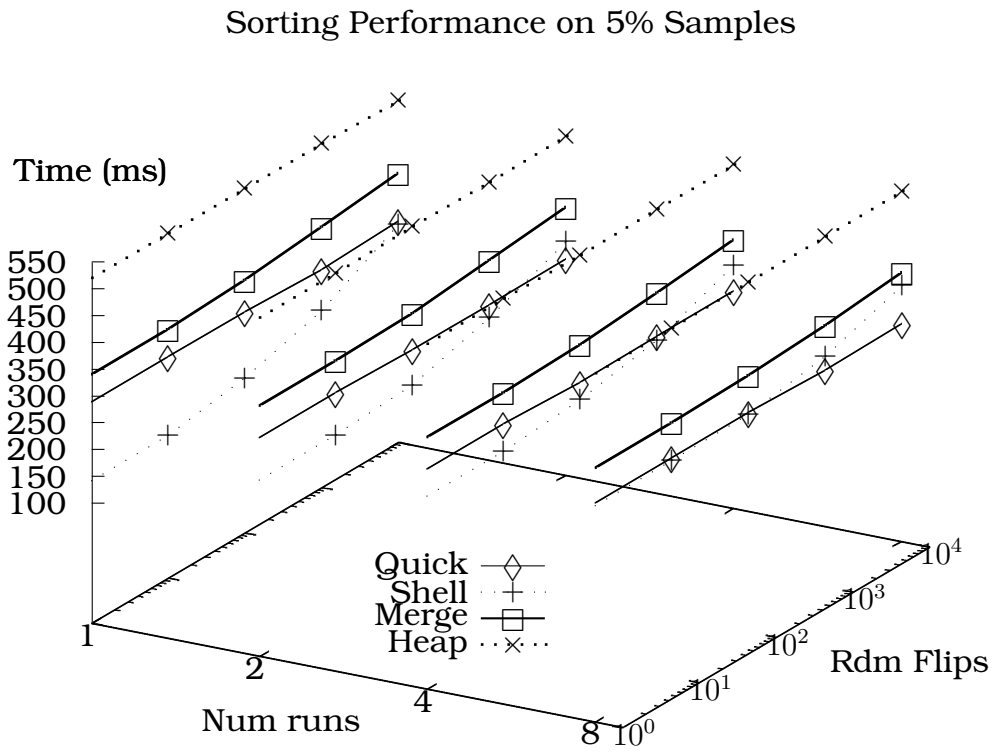


Figure 3.4: A comparison of quick sort, merge sort, Shell sort, and heap sort on permutations of size 5,000,000 with 1, 2, 4, and 8 runs at a 5% sample rate. The same pattern as in Figure 3.3.2 emerges. That is, Shell sort degrades as the various inputs become less and less sorted. The other relative rankings are similar as well as we see quick sort become the best performer while heap sort remains the worst performer.

In Figures 3.4, 3.5, and 3.6, we see the performance of each sorting algorithm across all ranges of problem specific features for different input proxy sizes. They reveal the ability of SLIP to produce proxies for sorting inputs whose performance reflects that of the original inputs. The larger proxies more accurately approximate the original inputs, but even the small samples of 5% do an excellent job of mimicking the originals.

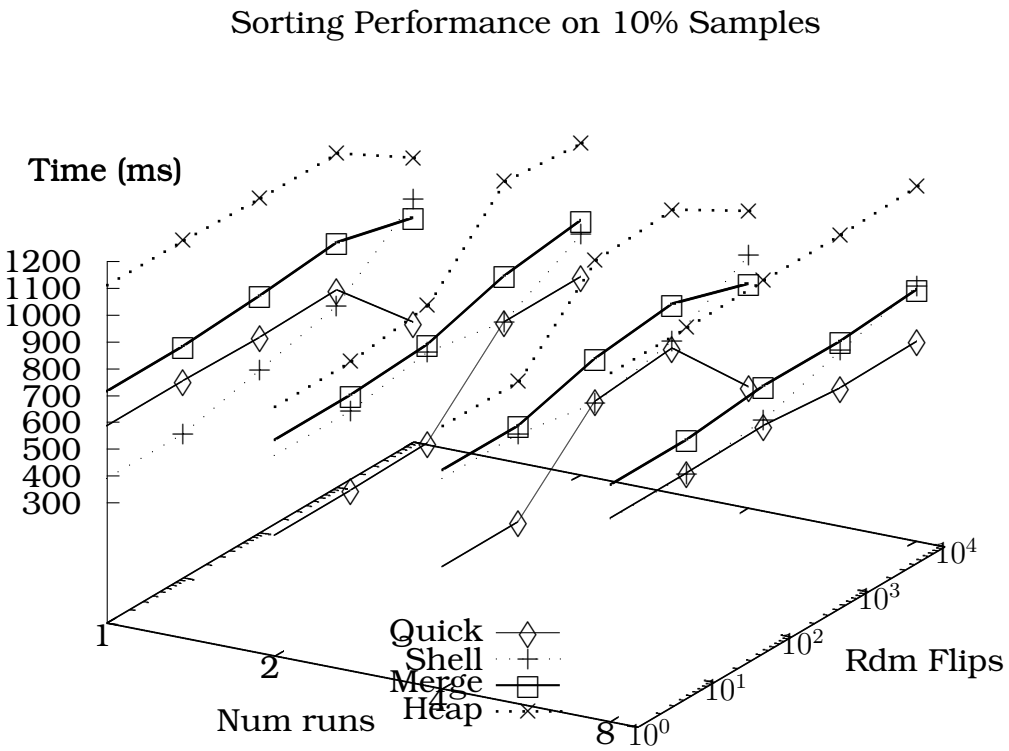


Figure 3.5: A comparison of quick sort, merge sort, Shell sort, and heap sort on permutations of size 5,000,000 with 1, 2, 4, and 8 runs at a 10% sample rate The performance of the algorithms match even more closely to the performance on the original inputs as shown in Figure 3.3.2 with the larger proxy size.

We will examine a subset of these experiments to see more precisely that the patterns that these proxies do indeed do an accurate job of predicting algorithmic performance on the larger inputs. In figures 3.7, 3.8, and 3.9, we see

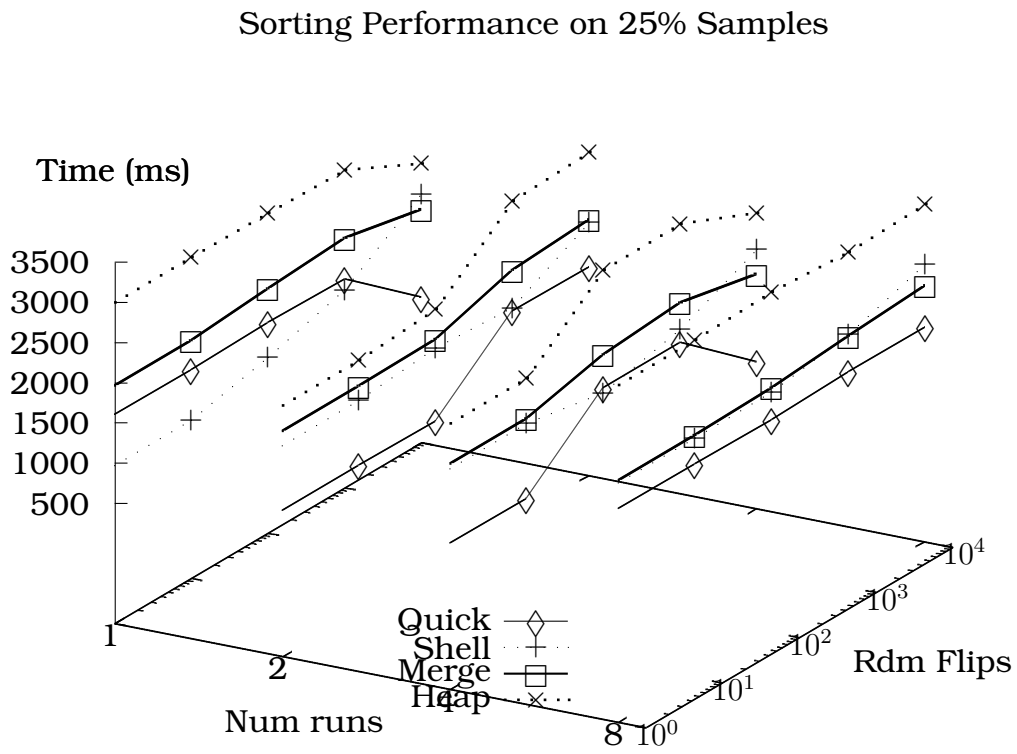


Figure 3.6: A comparison of quick sort, merge sort, Shell sort, and heap sort on permutations of size 5,000,000 with 1, 2, 4, and 8 runs at a 25% sample rate. Again, we see that the larger proxies more accurately reflect the performance of the algorithms on the original inputs.

performance of the different algorithms with proxies of different sample rates with one run inputs. The same patterns from the performance of the original inputs emerged – Shell sort is the most efficient when the inputs are sorted (or nearly sorted), but its performance degrades as inputs become more random. Other algorithms perform consistently across different inputs, but their relative rankings change. We also see that larger samples provide a more accurate reflection of the changes in these rankings for the original inputs.

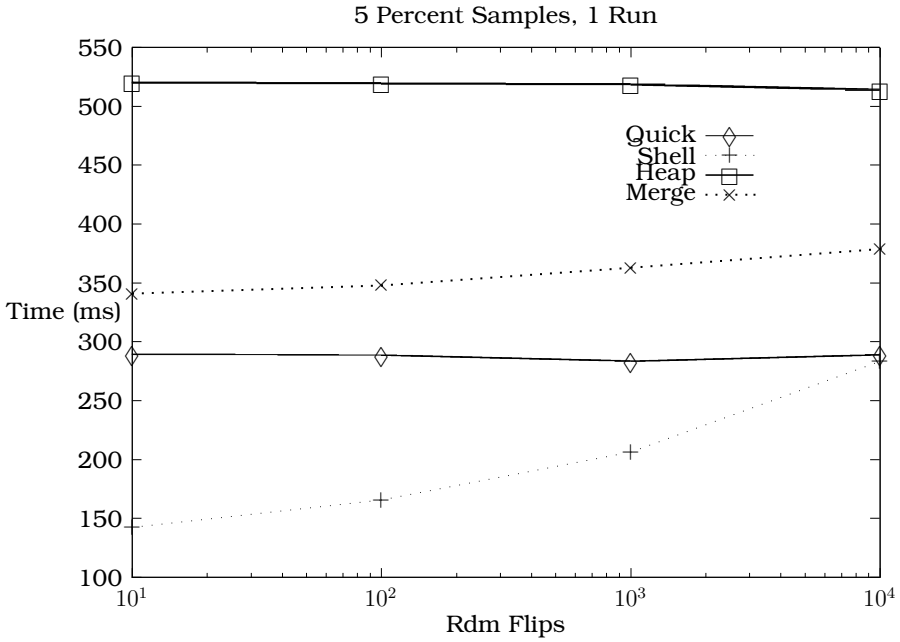


Figure 3.7: A comparison of performance on one run permutations at 5% sample rate.

In most cases, performance of a sorting algorithm on an in-order sample of a permutation proves to be an excellent indicator of sorting performance on the original input. For nearly sorted inputs, those algorithms which are optimal with respect to the number of inversions such as the Shell sort remain optimal because the sample reflects the structure of the input. That is, there are approximately the same number of inversions with respect to the size of the input. As

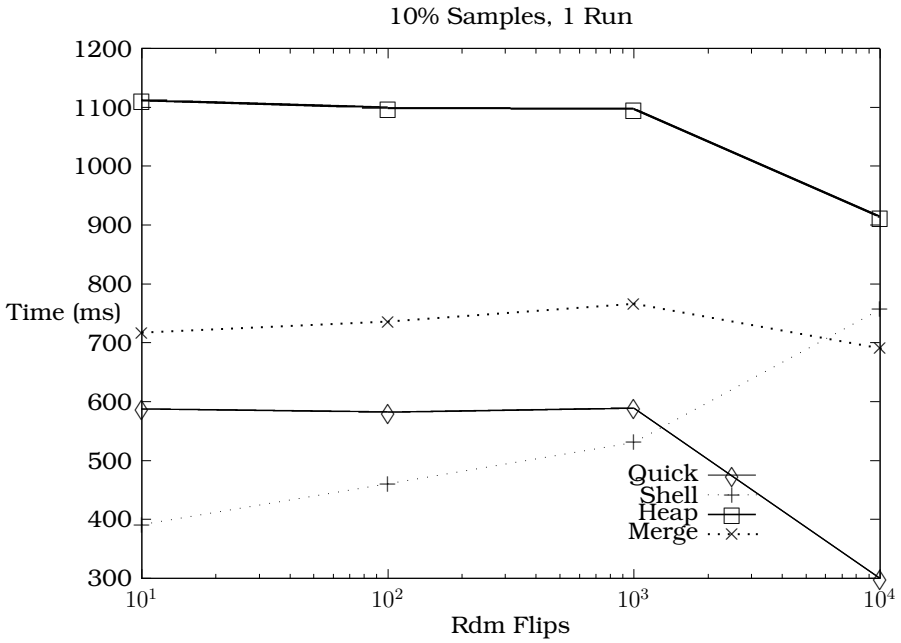


Figure 3.8: A comparison of performance on one run permutations at 10% sample rate.

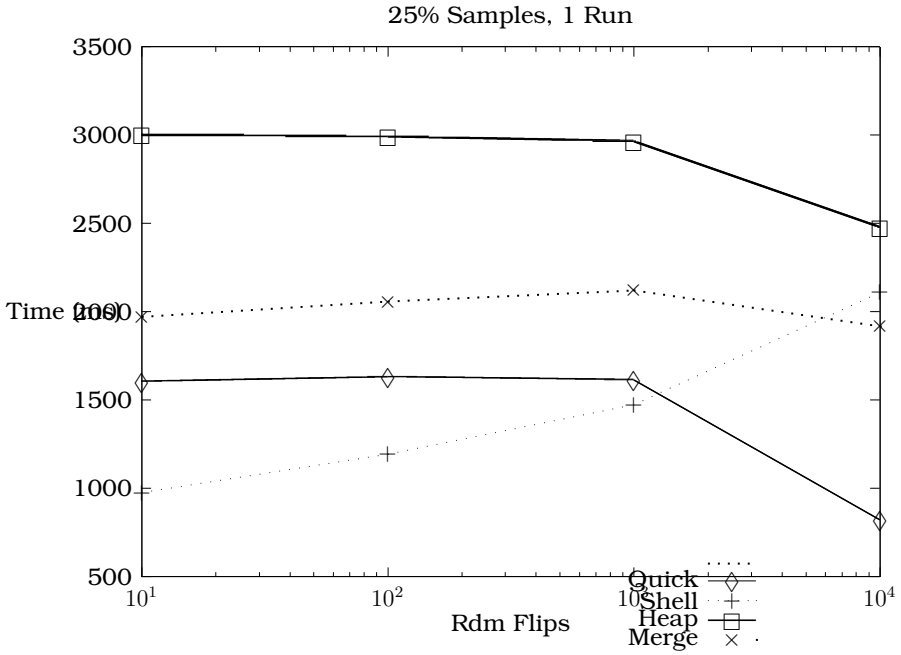


Figure 3.9: A comparison of performance on one run permutations at 25% sample rate.

the permutations grow more random, algorithms whose performance is not as strictly tied to particular aggregate features begin to outperform those algorithms more sensitive to them. This is also reflected in the performance on samples as the structure is maintained.

3.4 Selecting a Shell sort Increment Set

Shell sort is a family of sorting algorithms where the performance of different implementations depends almost exclusively on the selection of the increment set [38, 53]. The Shell sort acts as a multi-pass insertion sort where at the i^{th} pass, elements in the input which are incrementSet[i] distance apart are sorted. The final increment is 1, and so the final pass is a full insertion sort that compares all neighbors.

Much of the research has focused on finding an optimal increment set [53], but we use five increment sets that have different known efficiencies to test in a head-to-head comparison of performance. These include Shell, Knuth, Sedgewick, Hibbard, and Gonnet. The absolute and relative performance numbers are interesting for the Shell sort for several reasons. First, the overall numbers show that again the race on samples is a good indicator of the relative performance on the actual input. Like the previous experiments with four general sorting algorithms, the absolute numbers reflect expectations for performance for each increment set. Where there are mis-predictions in the rankings, it is usually at the milliseconds level of error. Also, the numbers show exactly when a bad increment set (Shell and Gonnet) actually can be useful. When an input is almost nearly sorted, the supposedly poorly performing Shell sorts are actually more efficient than the more optimal increment sets.

A final interesting issue the numbers bring up, though, is the performance of the bad increment sets when the permutation is transitioning from nearly sorted

to a more random input. At this point, the poorly performing Shell sorts actually perform well on the samples. This is caused by some features of the insertion sort and of the increment sets themselves. Where the inputs are between nearly sorted and random, insertion sort performs an order of magnitude better on the samples (scaled to the actual size of the input) because the samples are small enough that the percentage of random flips that make it through are not enough to send the performance into polynomial time. On the large input, however, the insertion sort degrades to polynomial time. The good increment sets avoid this degradation, though. Further research into this zone of degradation could yield interesting results with respect to optimal increment sets.

Similar to non-Shell sort algorithms, different increment sets cause the Shell sort to perform differently on different kinds of inputs. Because specific sets are more well-suited for different inputs, using sampling to test performance may provide insight into their relationship with problem specific features. In these experiments, the performance of Shell sort on large inputs as well as samples from them is compared using five different increment sets.

3.4.1 Experiments

Permutations of size 100,000 are used as inputs. Smaller inputs are used than the previous sorting experiments because the worst case for some of the increment sets (such as Shell and Gonnet) is extremely slow. Much larger inputs brings sorting to a halt for these two increment sets even though the other increment sets can handle them more gracefully. As in the previous sorting experiments, each input is randomly perturbed to provide a wide range of inputs from almost perfectly sorted to randomized with varying numbers of runs. Each input is sorted with Shell sort using the following well-known increment sets:

- Shell ($k/2$): This increment set begins at $log\ N$, and it is halved repeatedly. This is the basic increment set.
- Knuth ($(3^k) - 1)/2$): Each increment in this set is the result of this function where k is the element's index in the increment set.
- Sedgewick: This increment set contains the following elements – 0, 1, 5, 19, 41, 109, 209, 505, 929, 2161, 3905, 8929, 16001, 64769, 146305, 260609, 587521, and 1045505.
- Hibbard ($(2^k) - 1$): Each element in this increment set is one less than a power of two.
- Gonnet ($k/2.2$): Each increment in this set is simply the index of the element's position divided by the constant 2.2. This set represents an effort to improve on the original Shell increment set.

Then, random samples of 1, 5 and 10% of each input are sorted using the same increment sets. The results presented here are based on inputs of size 100,000 with a sample of 5%.

3.4.2 Results

The performance of different increment sets on the original inputs is presented in figure 3.10. In these experiments, we see more variation in prediction of performance for different increment sets. For example, the simplicity of the Shell and Gonnett increment sets allow the algorithms to excel on nearly sorted permutations as reflected in performance of the proxies. We see, however, that the performance predictions begin to fail for these particular increment sets as the inputs become more random. Even with the change from one to two runs, the Shell and Gonnet sets go from seconds to thousands of seconds.

		Time (ms)					Rank				
1 Run	Sorted	Shell	Knuth	Sedgewick	Hibbard	Gonnet	Shell	Knuth	Sedgewick	Hibbard	Gonnet
Sorted	Input	1288.0	2340.0	2950.0	2307.0	566.0	2	4	5	3	1
	Samp Avg	80.6	192.1	273.1	185.0	60.2	2	4	5	3	1
	Std Dev	5.9	13.6	15.2	3.7	11.5					
100	Input	2473.0	3131.0	3873.0	2970.0	2394.0	2	4	5	3	1
	Samp Avg	113.9	226.8	309.8	219.4	90.2	2	4	5	3	1
	Std Dev	12.8	13.2	19.1	14.7	9.7					
1000	Input	14417.0	4080.0	4238.0	3822.0	14488.0	4	2	3	1	5
	Samp Avg	224.5	275.5	356.6	261.5	207.9	2	4	5	3	1
	Std Dev	19.7	13.8	17.6	14.3	15.4					
2000	Input	26005.0	4225.0	5105.0	4003.0	26906.0	4	2	3	1	5
	Samp Avg	336.0	300.3	371.0	281.2	321.8	4	2	5	1	3
	Std Dev	18.1	11.3	17.8	11.9	22.7					
3000	Input	39637.0	4434.0	5296.0	4141.0	41287.0	4	2	3	1	5
	Samp Avg	429.5	315.1	383.5	291.2	419.6	5	2	3	1	4
	Std Dev	33.1	17.5	19.2	12.5	22.4					
4000	Input	50070.0	4828.0	4636.0	4367.0	50733.0	4	3	2	1	5
	Samp Avg	517.0	320.4	379.7	314.2	511.4	5	2	3	1	4
	Std Dev	29.0	9.3	15.2	48.7	22.4					
5000	Input	68838.0	4942.0	5378.0	4441.0	64692.0	5	2	3	1	4
	Samp Avg	621.4	330.9	385.8	308.4	615.5	5	2	3	1	4
	Std Dev	38.3	14.7	23.0	14.0	39.7					
10000	Input	137645.0	5502.0	4942.0	5273.0	132965.0	5	3	1	2	4
	Samp Avg	1144.9	368.4	420.9	341.4	1155.6	4	2	3	1	5
	Std Dev	68.8	24.0	56.8	25.2	64.3					

2 Runs	Sorted	Shell	Knuth	Sedgewick	Hibbard	Gonnet	Shell	Knuth	Sedgewick	Hibbard	Gonnet
Sorted	Input	3425269.0	3032.0	3676.0	5869.0	3231933.0	5	1	2	3	4
	Samp Avg	17221.4	255.0	346.1	314.0	17452.8	4	1	3	2	5
	Std Dev	567.1	14.9	13.1	13.4	577.1					
100	Input	3159905.0	3744.0	4254.0	6449.0	3232610.0	4	1	2	3	5
	Samp Avg	17226.0	282.1	370.2	337.4	17488.1	4	1	3	2	5
	Std Dev	505.8	17.2	22.8	16.8	510.8					
1000	Input	3185335.0	4296.0	4503.0	6936.0	3526836.0	4	1	2	3	5
	Samp Avg	17373.9	321.3	396.7	366.2	17607.8	4	1	3	2	5
	Std Dev	550.3	12.5	13.9	11.7	570.2					
2000	Input	3153525.0	4543.0	5406.0	7188.0	3534194.0	4	1	2	3	5
	Samp Avg	17209.4	338.4	408.9	380.2	17687.2	4	1	3	2	5
	Std Dev	554.9	14.9	17.6	14.9	671.0					
3000	Input	3129788.0	4854.0	5644.0	7457.0	3217149.0	4	1	2	3	5
	Samp Avg	17466.9	348.0	414.0	391.9	17711.5	4	1	3	2	5
	Std Dev	525.5	15.5	20.7	18.4	598.6					
4000	Input	3393631.0	4947.0	4842.0	7685.0	3260314.0	5	2	1	3	4
	Samp Avg	17713.0	369.7	434.8	410.7	17827.8	4	1	3	2	5
	Std Dev	713.0	23.3	55.8	25.1	565.7					
5000	Input	3691560.0	5040.0	5023.0	7717.0	3532517.0	5	2	1	3	4
	Samp Avg	17597.7	361.3	422.5	407.2	17855.7	4	1	3	2	5
	Std Dev	791.0	16.0	16.3	14.8	729.3					
10000	Input	3556984.0	5691.0	5431.0	8197.0	3559411.0	4	2	1	3	5
	Samp Avg	17943.2	388.6	437.7	432.4	18003.9	4	1	3	2	5
	Std Dev	727.5	21.5	18.8	16.5	808.9					

Figure 3.10: Summary of data from various Shell sort increment sets. As the inputs become more random, we see the Knuth and Sedgewick increment sets begin to outperform the Shell and Gonnet sets. With 2 runs in the input, we see that the Shell and Gonnet increment sets quickly become very bad solutions compared to other sets. The performance on the proxies reflect this.

For these nearly sorted inputs, though, the Gonnet does represent an improvement by many seconds over the Knuth, Sedgewick, and Hibbard sets, and this ranking is also reflected in the proxies. As soon as we get into the realm of the more unsorted, however, the contest comes down to the Knuth, Sedgewick, and Hibbard sets. With one run, the performance is virtually a tie with minor variations. When we transition to two runs, though, we see the Hibbard set begin to fall behind the others. In these cases, the Knuth and Sedgewick sets rise to the top of the rankings for both the original inputs and the proxies.

The difference in performance ranking at the 1000 flip point can be at least in part attributed to the size of the samples. Shell sort performs well on smaller inputs, and there appears to be an area where the original input is starting to become more random. Here, though, the sample contains few enough of the flips that the algorithms still perform relatively efficiently. As the climb toward randomness continues, more of the perturbations make it through the sample, and we see that the Shell and Gonnett increment sets sink to the bottom on both the original inputs as well as the samples.

As in the experiments using different sorting algorithms, the performance of the Shell sort is accurately reflected in its performance on a much smaller sample of that same input. Because insertion sort generally runs relatively more quickly on small inputs because of low overhead, the predictions do become inaccurate in a certain range of inputs. SLIP can be still be used effectively for inputs of various degrees of sortedness, but it also reveals interesting behavior for the different increment sets with different problem specific features.

3.5 Conclusions

We have shown that SLIP retains the problem specific features for inputs to the sorting problem for accurate approximation. In addition, these proxies may

be used as substitutes for the sorting algorithm and shell sort increment set selection problem. In particular, inputs to the sorting problem may be sampled to use as input for a race between sorting algorithms to accurately predict the most efficient solution for the original problem. Gains in performance do exist. In addition, this serves as an example of the ability of proxies to in fact provide accurate enough information about problem specific features of a traditional computational task. In the following chapters, applications of SLIP using graphs are explored which improve performance much more significantly.

Chapter 4

Classifying Large Graphs of Newsgroup Data

In mining information from very large graphs, processing time as well as system memory become computational bottlenecks as the properties of these large graphs must be compared through each iteration of an algorithm. This is a particularly pronounced problem for complex properties. For example, measures of distance between two inputs are used in many fundamental data mining algorithms including k-nearest neighbors for the classification task. Even the relatively efficient distance and similarity heuristics for large inputs, though, often require processing and memory well beyond linear with respect to the size of the input, and this rapidly becomes intractable with very large inputs. Complex properties such as the distance between two graphs can be extremely costly, but using proxies of these large graphs to calculate the same properties proves to reduce memory requirements and processing time significantly without sacrificing quality of classification. Because the vast amount of web data is easily and robustly represented with graphs, a data reduction technique that preserves the accuracy of mining algorithms on such inputs is important. We explore in this chapter the ability of SLIP to create significantly smaller yet equally expressive proxies for large graphs of web content, and we show how these proxies can be used to accurately classify web content more efficiently.

4.1 Introduction

Graphs provide a robust structure to model data for many computational problems. They are common inputs to problems in information systems, data mining, and image processing, and there is of course a rich theoretical background in graphs as well. Processing data in graph form can be expensive with respect to space and time efficiency, but very large graphs are not uncommon when modeling real world data. Graphs offer an especially rich representation for web-accessible content, and this chapter uses SLIP to improve the efficiency of graph classification created from such large real world corpora.

The data for the classification experiments comes from the collected text of twenty newsgroup postings available in the UC Irvine Knowledge Discovery in Databases Archive [54], but the techniques apply easily to any large collections of text such as those created from web sites or emails. After some preprocessing, each of the newsgroup graphs initially has over 20,000 nodes and several times that many edges, and they require a great deal of computing resources when used in even basic data mining tasks. We will show that SLIP can randomly sample these graphs in such a way that the proxies themselves may be used in the classification task. Because structural properties and problem specific features of the input are maintained, the use of the proxies improves the performance relative to both memory and speed without sacrifices in quality. Specifically, the k-nearest neighbors algorithm using significantly smaller graphs results in the same classifications when performed with the original inputs.

SLIP requires a means of randomly sampling the input as well as problem specific features which must be retained. For sampling, two methods are used including a hybrid node-edge selection technique as well as a random walk. The PSF for the classification task will be the distance between two graphs, and there are several ways to measure similarity between graphs. Various graph distance

metrics which can be used in common clustering and classification algorithms are listed, and results from experiments using these measures show that the classification of the proxies are of the same quality as the much larger original input graphs.

It is important to distinguish early between the effectiveness of the particular classification methods used and the effectiveness of the sampling method. This work is concerned with reproducing the same results with the proxies as with the much larger original graphs, and this means that misclassifications should be preserved as well. The focus is not necessarily on improving the accuracy of the classification method but rather on the ability of the proxy to replicate the accuracy of the original input. Any improvements in accuracy for a given method should transfer to the input proxies but with the advantage of significantly reduced processing.

4.2 Related Work

Sampling is widely used in computation, but it is typically associated with finding a representative subset of inputs from a large number of inputs. Sampling can also be used to reduce a single large input to a more manageable yet still representative structure for use in computations. Related work [30] shows the effectiveness of random sampling in reducing the space complexity of computational problems while maintaining the relevant structural features of large inputs. Such properties are useful in several areas. For example, the algorithm selection problem focuses on selecting the best algorithm among a collection of solutions whose efficiencies depend on features of the input [52]. In sorting, different sorting algorithms perform better or worse depending on the prior relative ordering of the input [36, 35]. For graphs, the efficiency of different algorithms

for finding the minimum spanning tree (for example, Kruskal's and Prim's algorithms) can vary according to the density of the graph.

Other related work examines the effectiveness in sampling in computation as well. Leskovec and Faloutsos test the ability of multiple sampling schemes to estimate individual features such as in- and out-degree of nodes from the original input [30]. They show that indeed sampling can be used to preserve context-independent features of many large graphs that model real world data such as citation networks. Other recent research using novel sampling techniques has been used to re-create the estimated coverage of a search engine's index [19]. The preservation of other graph properties from network graphs with relatively small samples is studied in [15].

There has also been work on using graphs in web content mining because of their ability to represent relationships between objects or elements in a set. This additional representational ability, though, often comes with a computational cost. Schenker et al. explore the use of graphs to overcome the inability of vector models to represent structural information [2]. They show that graphs are indeed effective structures for web content mining tasks such as classification and clustering, but the computational overhead can be prohibitive for large graphs. Other techniques for dimensionality reduction including latent semantic indexing [55] rely on the use of singular value decomposition to condense the input, and others have shown that such techniques can indeed provide as good or better performance for data mining tasks [29]. However, this work differs in that it attempts to maintain the rich structure inherent in graphs for use in classification.

Sampling of course has been used to choose from a large number of inputs to narrow the complexity of the input space in many data mining applications, but little work has focused on using sampling to reduce the complexity of a single

or many very large inputs to such a task. This research shows that sampling can also be used with large graphs to classify data more efficiently but just as accurately as with the original inputs. Using data from the text from a collection of twenty newsgroups postings [54], large graphs are created and classified into different groups using a traditional k-nearest neighbors algorithm using several distance metrics. Proxies of these large graphs are then processed in the same way, and results indicate samples indeed provide very accurate results with significantly smaller processing and memory requirements.

4.3 Sampling and Graphs

Large graphs as models of real world problems are a natural result of their ability to represent rich relationships between objects as well as a result of the constant increase in available machine-processable information. Processing such large graphs can become very costly, particularly for problems whose solutions scale polynomially or worse with the size of the input. Heuristics for reducing the search space are a common approach, and reducing the input to a more compact representation also can be critical in designing a tractable solution. Data reduction must not come at the expense of accuracy of the model, though. This section addresses accurately estimating such distance or similarity metrics for large graphs using SLIP.

4.3.1 Large Graphs and Their Features

Graphs are commonly defined as a structure of two sets – nodes and the edges that connect them. There are as many variations on this basic idea as there are problems to which to apply them, and these can be found in any introductory graph theory text. Graphs are commonly used not only to store objects

(nodes) but the relationships between them as well (edges). For any computational problem, there are problem specific features. Given a particular graph, such features may include the density or edge probability. These features can reveal a great deal about a particular input and its relationship with the problem. These PSF's can uniquely identify the input, and they can affect the performance of different algorithms used to solve the problem. They are also frequently the output of a problem themselves. For example, the similarity (or distance) between two inputs is required for many data mining processes such as clustering and classification. It is these distance metrics on which this chapter focuses.

Experiments on randomly generated graphs are useful to show that abstract features that are not context specific are maintained through sampling. Experiments on real data, however, show that features unique to a particular application or context can also be accurately estimated with an appropriate sampling technique. Large volumes of data naturally result in large computational representations, and these structures are rich in features as well as noise. Data to create such large graphs are readily available in various domains such as network traffic analysis, image processing, and the mining of data from large text corpora [2], and this work uses sampling to filter noise out of large graphs created from web content for more efficient classification.

4.3.2 Sampling Graphs

There are a variety of ways to sample graphs. While some focus on the set of nodes, others focus on the edge set. Random walks and its variations are also often used to explore graphs to create a sample. Each method has its own advantages and disadvantages, and the best method often depends on the application or data itself. In random node selection, random nodes are selected until a target sample size is reached. Variations often differ on exactly which edges to include

from a node's adjacency list. In random edge selection, random edges are added to a target sample's edge list until a required size is reached. Both of these techniques have their biases toward or against certain kinds of nodes, though.

Random walk-based sampling techniques often select a random node from which to start a walk across a graph by selecting edges to traverse from the current node's adjacency list. Variations often manipulate various probabilities that influence the chance of returning to the start node or a new random node, for example. In this chapter, experiments use two sampling techniques – a hybrid edge-node sampling scheme as well as a random walk. The details of each are are discussed in Section 4.5. The ability of these sampling techniques to create proxies which approximate problem specific features such as the relative distance between graphs will be evaluated for the classification task in this chapter, and the same methods will be used for the clustering task in the next chapter.

4.4 Graph Similarity Measures

Measuring how alike two graphs are is a computationally complex problem. If two graphs are considered isomorphic, we might say that they are the same graph with different labels. If they are not exactly isomorphic, though, we might want to describe how close they are to being exactly alike (or unalike). Such notions of relative proximity are very useful in tasks such as classification and clustering. Though we ideally want a highly accurate but computationally inexpensive distance metric, there exists as always a tradeoff. Several such measures are reviewed and proposed in [2]. These experiments use five particular metrics for similarity calculations on the original graphs as well as samples taken from them.

4.4.1 Graph Distance Metrics

Measuring the similarity between graphs provides a notion of how far apart or how close two graphs are, but they can be most useful when several such graphs are being compared. Schenker lists several such graph similarity techniques including but not limited to:

- Graph and subgraph isomorphism
- Graph edit distance
- Probabilistic
- Distance preservation
- Maximum common subgraph
- Minimum common supergraph

Two graphs are said to be isomorphic when there exists a 1-to-1 correspondence between the nodes of two graphs such that edges and labels are preserved while subgraph isomorphism requires such a mapping of a graph to a subgraph of another [46, 48]. These are known difficult problems, and Ullman provides an algorithm for subgraph isomorphism [56]. With isomorphic measures of similarity, however, the results are absolute. Either graphs are exactly the same or they are not. Schenker and others define four properties for a graph similarity measure [48, 49, 50, 47]:

- Boundary condition: $d(G_1, G_2) \geq 0$
- Isomorphic graphs have zero distance: $d(G_1, G_2) = 0 \rightarrow G_1 \cong G_2$

- Symmetry: $d(G_1, G_2) = d(G_2, G_1)$
- Triangle inequality: $d(G_1, G_3) \geq d(G_1, G_2) + d(G_2, G_3)$

Edit distances measure the cost of transforming one structure into another [40, 39, 57]. For graphs, we may define such operations as adding, deleting, and substituting nodes and edges from a graph to transform it into another. If each operation has a cost associated with it, we can measure the total cost by summing each operation [46]. With this many parameters, though, graph edit distances can be application specific [44].

Probabilistic approaches attempt to match a graph with a model graph based on matching super-cliques of the input graph with super-cliques of the model graph [43]. Relying on Bayesian techniques and other probability theory, Wilson and Hancock have created such a description. It has been refined to only require one parameter. Distance preservation approaches focus on maintaining the minimum distance between two nodes in two graphs [45]. Such an approach can require the calculation of distances between all pairs of nodes in the input graphs.

Experiments in this chapter and the next use variations of the maximum common subgraph and minimum common supergraph distance measures. In addition, a new technique using graph sampling to determine similarity is introduced.

4.4.2 Maximum Common Subgraph

The maximum common subgraph technique attempts to find the largest subgraph that two graphs have in common, and it is related to the graph edit distance [46]. Given graphs G_1 and G_2, we define *mcs(G_1, G_2)* as the maximum common subgraph of G_1 and G_2 if it is a subgraph of G_1, it is a subgraph of G_2,

and there exists no other subgraph that is contained in both G_1 and G_2 that is larger. With unlabeled graphs, such a measure would be very inefficient because of the exponential number of possible pairings. With labeled graphs, however, the maximum common subgraph is computationally manageable because the labels significantly reduce the possible matchings. Even if labelled, though, space and time requirements are still very high with very large graphs. The algorithm used in the experiments is as follows:

Algorithm 1: Maximum Common Subgraph

```
Input: Graphs G1 and G2
Output: Maximum Common Subgraph Gmcs of G1 and G2
for edge e ∈ G1 do
    if e ∈ G2 then
        add e to Gmcs
    end
end
return Gmcs
```

Distance measures used based on the maximum common subgraph include the following:

$$d_{mcs}(G_1, G_2) = 1 - \frac{|mcs(G_1, G_2)|}{max(|G_1|, |G_2|)} \tag{4.1}$$

$$d_{wgu}(G_1, G_2) = 1 - \frac{|mcs(G_1, G_2)|}{|G_1| + |G_2| - |mcs(G_1, G_2)|)} \tag{4.2}$$

$$d_{ugu}(G_1, G_2) = |G_1| + |G_2| - 2|mcs(G_1, G_2)| \tag{4.3}$$

The WGU metric in Formula 4.2 considers the union of the two graphs in its metric [49]. By summing the two graph sizes and removing the size of their intersection, the measure allows for both graphs (not just the largest) to play

a role in the distance calculation. The UGU metric in Formula 4.3 is a non-normalized measure of distance that also considers the union of the graphs [46]. By subtracting twice the size of the intersection, two exactly matching graphs should have a distance of zero. However, because the distance is not set to an interval such as $[0, 1]$, we will see that significantly different sized graphs can adversely affect the ability of algorithms such as k-means to classify effectively.

4.4.3 Minimum Common Supergraph

The minimum common supergraph attempts to find the smallest supergraph of two graphs, and it is related to the union of two graphs. Given graphs G_1 and G_2, we define *MCS(G_1, G_2)* as the minimum common supergraph of G_1 and G_2 if it is a supergraph of G_1, it is a supergraph of G_2, and there exists no smaller supergraph of both [48]. Space and time requirements are still very high with very large graphs, however. The algorithm used in the experiments is as follows:

Algorithm 2: Minimum Common Supergraph

Input: Graphs G_1 and G_2
Output: Minimum Common Supergraph G_{MCS} of G_1 and G_2
for *each edge* $e \in G_1$ **do**
    add e to G_{MCS}
end
for *edge* $e \in G_2$ **do**
    if $e \notin G_{MCS}$ **then**
        add e toG_{MCS}
    end
end
return G_{MCS}

The distance measure used based on the minimum common supergraph is defined as:

$$d_{MCS}(G_1, G_2) = 1 - \frac{|mcs(G_1, G_2)|}{max(|MCS(G_1, G_2)|)} \tag{4.4}$$

The metric in Formula 4.4 uses the maximum common subgraph as a lower bounds of sort on the similarity between two graphs while the minimum common subgraph represents the upper bound of similarity [50, 2].

4.4.4 Sample Intersection Distance

A new distance measure introduced in this chapter relies on the number of common edges taken in a random sample of edges from one of two input graphs. If a sample of size n is taken, the distance is calculated as one minus the ratio of number of randomly chosen edges that exist in both graphs to n. If n edges are chosen, and all n exist in both graphs, then the distance is considered to be zero. This method is meant to be a quickly computed estimate of the maximum common subgraph measure. That is, it only considers edges (with the same source and target nodes) that the graphs have in common, but it does not rely on an exhaustive search for all such edges. Results show that this measure proves to be a good distance metric.

Algorithm 3: Sample Intersection

```
Input: Graphs G_1 and G_2, sample size n
Output: Number of edges which exist in both graphs
while edges sampled < n do
    select random edge e from G_1
    if e ∈ G_1 AND e ∈ G_2 then
        increment match count m
    end
end
return m
```

The distance measure used based on the sample intersection distance is defined as:

$$d_{sid}(G_1, G_2) = 1 - \frac{m}{n} \tag{4.5}$$

where n is the number of random edges selected and m is the number of these edges found to be in both input graphs.

With large graphs, these calculations can rapidly exceed the computational limits of a particular application. This can of course depend on the efficiency of the underlying data structures, but SLIP provides an implementation-independent way to reduce the size of the inputs significantly while still preserving its aggregate structure so that relative ordering with respect to similarity between all pairs of graphs is maintained given a set of inputs. Because the distance metrics are directly tied to the number of edges in one or both input graphs (with some multiplicative factor based on the graph data structure efficiency), any reduction in the number of edges in the proxy should result in large gains in performance for data mining tasks with little or no sacrifice in quality. Details about results from all of the experiments using the various sampling techniques and distance measures are provided in the following sections.

4.5 Sampling for Similarity

The effectiveness of graph sampling methods is often dependent on the data being represented. We examine in detail a hybrid random edge-node selection method for creating a subgraph of an input, and we also consider a random walk-based sampling method. In order to avoid introducing new information into any samples, we simply need to avoid creating new edges between nodes. The more difficult part of the sampling is maintaining the general structure for a graph for different kinds of graphs.

4.5.1 Hybrid Edge-Node Random Sampling

A randomly selected subgraph *g'* of size *s* of some graph *G* can be created by randomly selecting *s* nodes and adding them to *g'*. Variations might select all or

part of each node's adjacency list to add to *g'* as well. Alternatively, edges can be randomly selected from *G's* edge list to add to *g'* along with its source and destination nodes. Random node or edge selection to create samples has certain biases toward or against nodes with higher or lower in and out degrees, and some of these can be avoided through a hybrid approach. The following algorithm creates a subgraph *g'* of *G* of size *s* by randomly selecting edges, adding nodes, and then processing the adjacency lists of each node previously added.

Algorithm 4: Hybrid Random Node-Edge Graph Sampling

```
Input: Graph G, Subgraph proxy size s
Output: Subgraph g' of G
while | g' | ≤ s do
    select random edge e from G
    if e ∉ g' then
        if source node ∉ g' then
            add source node to g'
        end
        if destination node ∉ g' then
            add destination node to g'
        end
    end
end
for all nodes n ∈ g' do
    for edge e ∈ n's adjacency list do
        if destination node ∉ g' then
            remove e
        end
    end
end
return g'
```

A simple random selection of nodes can lead to a graph that ignores the importance that "hub" nodes play in graphs. When only nodes are considered, all nodes stand an equal chance of selection and so nodes with many edges do not have any advantage. This can ignore a major feature of a graph. On the other hand, simply selecting edges can lead to similar but different problems. Consider, for example, a fully connected graph. Randomly selecting half of the edges

for inclusion in the sample will likely include every node in the sample but many fewer edges. Fundamental properties of the graph change such as its relative density, and there is no way to get a sample that also represents a full connected graph. This method thus has the same problems as the first sampling method – the structure and features of the input are lost.

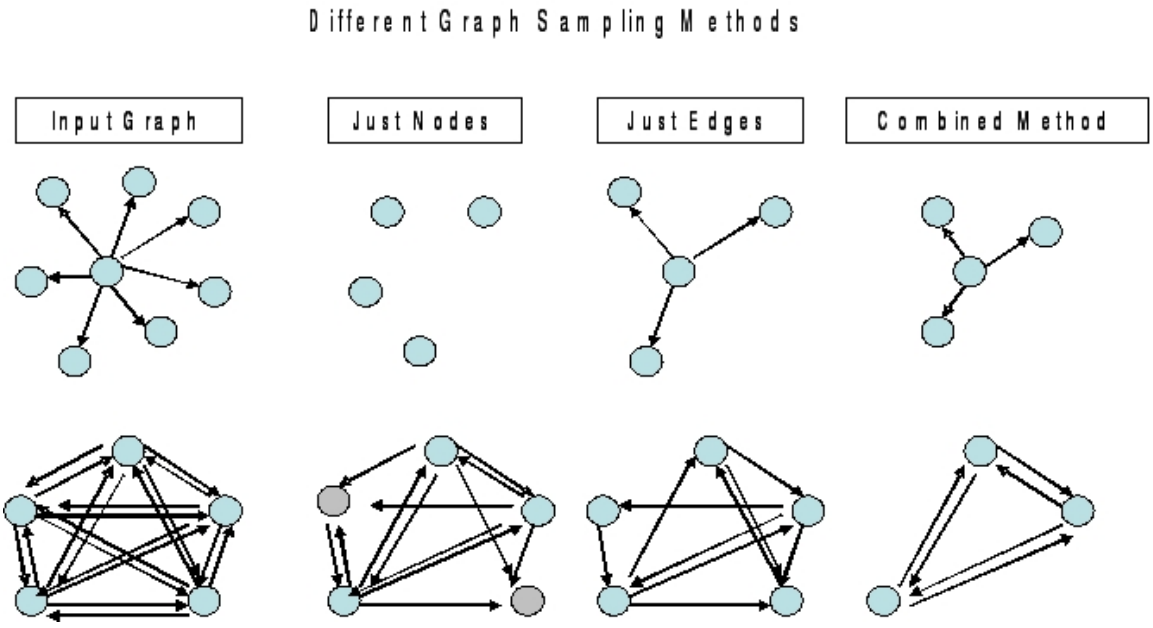


Figure 4.1: Flexibility of the Hybrid Graph Sampling Technique to Preserve Structure.

A visual comparison of node, edge, and the hybrid technique is shown in figure 4.1. The hybrid method not only allows for making the chance of a node being in the sample proportional to the number of edges in which it participates, but it also limits the number of nodes in the sample while maintaining certain structure from the input. This sampling method generalizes well for many types of graphs to maintain general structure. Particular data may lend itself well to other techniques such as random walk sampling. For example, an application may require that the sample be a connected component of the original input. In these experiments, however, no such requirements exist.

4.5.2 Random Walk Sampling

A subgraph *g'* of size *s* of some graph *G* can be created through a random walk. By starting at some randomly chosen node, a walk is performed where new edges are added to the sample if it has not been encountered before. A maximum path length from the starting node may be set, and a probability associated with re-starting at the same original node or a new randomly chosen one is another parameter that affects the type of sample created. The process ends when the sample reaches a required size.

Algorithm 5: Random Walk Graph Sampling

Input: Graph G, sample size s, max path length l, new start node probability p
Output: Subgraph g' of G
select random edge e from G
set source node of e as current node n
add n to g'
while $\mid g' \mid \leq s$ **do**
    reset step count
    while $stepnumber < l$ **do**
        select random edge e originating from n
        if $e \notin g'$ **then**
            add e to g'
            add target node of e to g'
            set n to target node
        end
        increment step count
    end
    with probability p select new start node
end
return g'

Both the hybrid node-edge selection technique as well as the random walk are used in experiments to test whether samples retain the aggregate property of distance between two graphs as measured by the mcs, MCS, WGU, UGU, and SID distance metrics. The random walk samples in these experiments set the probability of a re-starting at the previous start node at 0.9 with a maximum

path length of 20. These are parameters that are domain specific, but a thorough examination of these parameters for different types of data is outside the scope of this chapter. Similar values are used in other work using random walks for sampling graphs [30].

4.5.3 DATA

Randomly generated artificial graphs provide complete control over features of the graph at the individual node and edge level as well as the aggregate level, but graphs representing real world data provide motivation for application of sampling techniques [58, 30]. A collection of newsgroup postings from the UC-Irvine Knowledge Discovery in Databases archive is used [54]. These files contain 20,000 messages from twenty newsgroups (1,000 from each newsgroup). The groups are listed in Table 4.1:

Atheism	Politics - Misc	Windows X	Autos
Religion	Computer Graphics	Cryptology	Motorcycles
Christianity	IBM Hardware	Electronics	For Sale
Politics - Guns	Mac Hardware	Medicine	Baseball
Politics - Mideast	Windows Misc	Space	Hockey

Table 4.1: Categories of Newsgroups from UC-Irvine Repository

From these, twenty graphs are constructed based on the words contained in the messages and their proximity to each other. Stop word removal is performed to remove many common words from all inputs. This is a common technique when processing text input to reduce noise by removing elements which do not help to distinguish or define the input. Other preprocessing techniques such as stemming are possible, but some noise is left in the graph representation by performing only basic stop word removal to test how robust the sampling techniques

can be. Several measures could be taken to improve the distance measures and the consequent classification, but the goal of these experiments it to discover if samples make the *same* predictions as the original inputs (not whether the original predictions are the most accurate possible). Directed edges are created between words within three positions of each other, but this parameter can be adjusted to create more or less dense graphs. Sample sizes of 15, 30, and 50% are used in experiments.

4.5.4 Experiments

Each distance metric is used to calculate the similarity between all pairs of input graphs as well as pairs of the proxy graphs. These proxies are created using both the hybrid node-edge sampling technique as well as the random walk method. Distance data are provided for all metrics for the original inputs, but only the maximum common subgraph distance data are provided for the 15, 30, and 50% proxies. Exhaustive distance information for all metrics and sample sizes may be found in Appendix A. Specific examples analyzing the ability of the proxies to predict graph similarity are provided along with the raw data.

Original Inputs

In tables 4.2, 4.3, and 4.4, we see the distances on the original inputs using the maximum common subgraph, WGU, and UGU metrics. All of these metrics rely on the size of the maximum common subgraph as seen in formulas 4.1, 4.2, and 4.3.

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	**0.7308**	0.9227	0.9332	0.9470	0.9338	0.9548	0.9578	0.9567	0.9683	0.9637	0.9416	0.9526	0.9469	0.9486	0.9514	0.9552	0.9655	0.9519	0.9539
Religion	**0.7308**	0.0000	0.9154	0.8663	0.9310	0.7769	0.9549	0.9574	0.9568	0.9643	0.9612	0.9391	0.9542	0.9269	0.9142	0.9514	0.9559	0.9637	0.9537	0.9586
Christian	0.9227	**0.9154**	0.0000	0.9404	0.9480	0.9380	0.9552	0.9593	0.9589	0.9689	0.9636	0.9463	0.9562	0.9465	0.9527	0.9536	0.9606	0.9655	0.9554	0.9584
Guns	0.9332	0.8663	0.9404	0.0000	0.9191	**0.8380**	0.9488	0.9535	0.9517	0.9641	0.9585	0.9127	0.9496	0.9250	0.9284	0.9401	0.9473	0.9579	0.9466	0.9500
Mideast	0.9470	0.9310	0.9480	0.9191	0.0000	**0.8482**	0.9629	0.9681	0.9658	0.9697	0.9667	0.9495	0.9658	0.9417	0.9431	0.9608	0.9657	0.9721	0.9614	0.9619
Misc Politics	0.9338	**0.7769**	0.9380	0.8380	0.8482	0.0000	0.9546	0.9574	0.9555	0.9614	0.9582	0.9316	0.9542	0.9267	0.9290	0.9471	0.9543	0.9615	0.9501	0.9516
Graphics	0.9548	0.9549	0.9552	0.9488	0.9629	0.9546	0.0000	0.9239	0.9298	0.9391	**0.9151**	0.9377	0.9349	0.9388	0.9360	0.9468	0.9546	0.9407	0.9538	0.9533
IBM Hardware	0.9578	0.9574	0.9593	0.9535	0.9681	0.9574	0.9239	0.0000	**0.8910**	0.9351	0.9418	0.9469	0.9115	0.9497	0.9517	0.9320	0.9428	0.9180	0.9438	0.9526
Mac Hardware	0.9567	0.9568	0.9589	0.9517	0.9658	0.9555	0.9298	**0.8910**	0.0000	0.9507	0.9456	0.9476	0.9174	0.9503	0.9498	0.9338	0.9388	0.9200	0.9426	0.9551
Windows Misc	0.9683	0.9643	0.9689	0.9641	0.9697	0.9614	0.9391	**0.9351**	0.9507	0.0000	0.9473	0.9592	0.9566	0.9600	0.9615	0.9612	0.9674	0.9576	0.9671	0.9689
Windows X	0.9637	0.9612	0.9636	0.9585	0.9667	0.9582	**0.9151**	0.9418	0.9456	0.9473	0.0000	0.9469	0.9486	0.9515	0.9507	0.9574	0.9622	0.9552	0.9626	0.9640
Cryptography	0.9416	0.9391	0.9463	**0.9127**	0.9495	0.9316	0.9377	0.9469	0.9476	0.9592	0.9469	0.0000	0.9417	0.9407	0.9380	0.9476	0.9542	0.9562	0.9513	0.9564
Electronics	0.9526	0.9542	0.9562	0.9496	0.9658	0.9542	0.9349	**0.9115**	0.9174	0.9566	0.9486	0.9417	0.0000	0.9453	0.9440	0.9240	0.9371	0.9201	0.9428	0.9557
Medicine	0.9469	0.9269	0.9465	0.9250	0.9417	0.9267	0.9388	0.9497	0.9503	0.9600	0.9515	0.9407	0.9453	0.0000	**0.9176**	0.9459	0.9529	0.9557	0.9525	0.9528
Space	0.9486	**0.9142**	0.9527	0.9284	0.9431	0.9290	0.9360	0.9517	0.9498	0.9615	0.9507	0.9380	0.9440	0.9176	0.0000	0.9502	0.9559	0.9567	0.9556	0.9563
Autos	0.9514	0.9514	0.9536	0.9401	0.9608	0.9471	0.9468	0.9320	0.9338	0.9612	0.9574	0.9476	0.9240	0.9459	0.9502	0.0000	**0.9214**	0.9260	0.9381	0.9503
Motorcycles	0.9552	0.9559	0.9606	0.9473	0.9657	0.9543	0.9546	0.9428	0.9388	0.9674	0.9622	0.9542	0.9371	0.9529	0.9559	**0.9214**	0.0000	0.9455	0.9432	0.9572
For Sale	0.9655	0.9637	0.9655	0.9579	0.9721	0.9615	0.9407	**0.9180**	0.9200	0.9576	0.9552	0.9562	0.9201	0.9557	0.9567	0.9260	0.9455	0.0000	0.9486	0.9551
Baseball	0.9519	0.9537	0.9554	0.9466	0.9614	0.9501	0.9538	0.9438	0.9426	0.9671	0.9626	0.9513	0.9428	0.9525	0.9556	0.9381	0.9432	0.9486	0.0000	**0.9313**
Hockey	0.9539	0.9586	0.9584	0.9500	0.9619	0.9516	0.9533	0.9526	0.9551	0.9689	0.9640	0.9564	0.9557	0.9528	0.9563	0.9503	0.9572	0.9551	**0.9313**	0.0000

Table 4.2: Distance between original input using maximum common subgraph. The closest graph for each is row is in bold.

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.8357	0.9586	0.9647	0.9692	0.9620	0.9766	0.9761	0.9747	0.9811	0.9798	0.9691	0.9736	0.9712	0.9721	0.9731	0.9743	0.9807	0.9730	0.9763
Religion	0.8357	0.0000	0.9549	0.9264	0.9614	0.8664	0.9760	0.9748	0.9737	0.9796	0.9794	0.9678	0.9733	0.9618	0.9549	0.9719	0.9736	0.9788	0.9729	0.9777
Christian	0.9586	0.9549	0.0000	0.9691	0.9706	0.9655	0.9767	0.9764	0.9754	0.9820	0.9803	0.9723	0.9749	0.9717	0.9751	0.9737	0.9769	0.9802	0.9743	0.9780
Guns	0.9647	0.9264	0.9691	0.0000	0.9533	0.9038	0.9734	0.9731	0.9711	0.9790	0.9773	0.9542	0.9712	0.9597	0.9616	0.9659	0.9690	0.9759	0.9693	0.9736
Mideast	0.9692	0.9614	0.9706	0.9533	0.0000	0.9160	0.9789	0.9800	0.9780	0.9839	0.9824	0.9714	0.9789	0.9678	0.9686	0.9759	0.9782	0.9827	0.9759	0.9779
Misc Politics	0.9620	0.8664	0.9655	0.9038	0.9160	0.0000	0.9745	0.9736	0.9716	0.9790	0.9783	0.9617	0.9719	0.9600	0.9613	0.9678	0.9713	0.9764	0.9693	0.9723
Graphics	0.9766	0.9760	0.9767	0.9734	0.9789	0.9745	0.0000	0.9557	0.9580	0.9634	0.9520	0.9673	0.9629	0.9669	0.9655	0.9702	0.9737	0.9661	0.9738	0.9757
IBM Hardware	0.9761	0.9748	0.9764	0.9731	0.9800	0.9736	0.9557	0.0000	0.9402	0.9570	0.9641	0.9691	0.9529	0.9700	0.9712	0.9638	0.9701	0.9569	0.9708	0.9733
Mac Hardware	0.9747	0.9737	0.9754	0.9711	0.9780	0.9716	0.9580	0.9402	0.0000	0.9669	0.9655	0.9685	0.9546	0.9695	0.9692	0.9635	0.9677	0.9564	0.9690	0.9739
Windows Misc	0.9811	0.9796	0.9820	0.9790	0.9839	0.9790	0.9634	0.9570	0.9669	0.0000	0.9707	0.9761	0.9720	0.9772	0.9781	0.9753	0.9787	0.9725	0.9788	0.9814
Windows X	0.9798	0.9794	0.9803	0.9773	0.9824	0.9783	0.9520	0.9641	0.9655	0.9707	0.0000	0.9709	0.9688	0.9743	0.9738	0.9745	0.9766	0.9728	0.9774	0.9799
Cryptography	0.9691	0.9678	0.9723	0.9542	0.9714	0.9617	0.9673	0.9691	0.9685	0.9761	0.9709	0.0000	0.9664	0.9685	0.9671	0.9703	0.9731	0.9748	0.9720	0.9769
Electronics	0.9736	0.9733	0.9749	0.9712	0.9789	0.9719	0.9629	0.9529	0.9546	0.9720	0.9688	0.9664	0.0000	0.9677	0.9669	0.9600	0.9665	0.9580	0.9704	0.9755
Medicine	0.9712	0.9618	0.9717	0.9597	0.9678	0.9600	0.9669	0.9700	0.9695	0.9772	0.9743	0.9685	0.9677	0.0000	0.9570	0.9684	0.9716	0.9738	0.9720	0.9742
Space	0.9721	0.9549	0.9751	0.9616	0.9686	0.9613	0.9655	0.9712	0.9692	0.9781	0.9738	0.9671	0.9669	0.9570	0.0000	0.9710	0.9735	0.9745	0.9739	0.9763
Autos	0.9731	0.9719	0.9737	0.9659	0.9759	0.9678	0.9702	0.9638	0.9635	0.9753	0.9745	0.9703	0.9600	0.9684	0.9710	0.0000	0.9573	0.9608	0.9675	0.9727
Motorcycles	0.9743	0.9736	0.9769	0.9690	0.9782	0.9713	0.9737	0.9701	0.9677	0.9787	0.9766	0.9731	0.9665	0.9716	0.9735	0.9573	0.0000	0.9713	0.9700	0.9757
For Sale	0.9807	0.9788	0.9802	0.9759	0.9827	0.9764	0.9661	0.9569	0.9564	0.9725	0.9728	0.9748	0.9580	0.9738	0.9745	0.9608	0.9713	0.0000	0.9735	0.9749
Baseball	0.9730	0.9729	0.9743	0.9693	0.9759	0.9693	0.9738	0.9708	0.9690	0.9788	0.9774	0.9720	0.9704	0.9720	0.9739	0.9675	0.9700	0.9735	0.0000	0.9613
Hockey	0.9763	0.9777	0.9780	0.9736	0.9779	0.9723	0.9757	0.9733	0.9739	0.9814	0.9799	0.9769	0.9755	0.9742	0.9763	0.9727	0.9757	0.9749	0.9613	0.0000

Table 4.3: Distance between original Input using WGU metric. The relative distances are very similar to the maximum common subgraph distances.

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0	55589	69855	70228	79275	76302	71098	63643	61476	85435	77761	71118	64241	73667	73722	64869	62607	64675	63842	69760
Religion	55589	0	72786	68287	81532	65793	74605	67058	64925	88792	81310	74471	67780	75786	74667	68272	66092	68040	67395	73563
Christian	69855	72786	0	72929	81582	78889	73213	65778	63663	87708	79962	73669	66518	75834	76263	67034	65034	66730	66107	72111
Guns	70228	68287	72929	0	78321	69274	72248	64859	62636	86691	78995	70564	65537	73555	73752	65523	63537	65669	64964	70996
Mideast	79275	81532	81582	78321	0	78537	81611	74348	72047	96204	88430	81481	75114	83132	83175	75352	73272	75194	74345	80163
Misc Politics	76302	65793	78889	69274	78537	0	79008	71517	69256	93359	85805	78072	72201	80013	80132	72281	70385	72357	71490	77388
Graphics	71098	74605	73213	72248	81611	79008	0	61843	60202	83225	74295	71636	63651	73813	73510	65265	63305	63573	64724	70458
IBM Hardware	63643	67058	65778	64859	74348	71517	61843	0	51219	75056	68906	64599	55334	66962	67045	57222	55552	55306	57025	62783
Mac Hardware	61476	64925	63663	62636	72047	69256	60202	51219	0	74595	67147	62566	53615	64931	64808	55249	53331	53341	54864	60878
Windows Misc	85435	88792	87708	86691	96204	93359	83225	75056	74595	0	90542	86455	78264	88928	88997	79468	77562	77856	79009	84967
Windows X	77761	81310	79962	78995	88430	85805	74295	68906	67147	90542	0	78251	70480	81046	80881	71974	69850	70554	71373	77265
Cryptography	71118	74471	73669	70564	81481	78072	71636	64599	62566	86455	78251	0	65175	75123	74824	66355	64323	65791	65576	71736
Electronics	64241	67780	66518	65537	75114	72201	63651	55334	53615	78264	70480	65175	0	67578	67389	57696	56068	56334	57899	63987
Medicine	73667	75786	75834	73555	83132	80013	73813	66962	64931	88928	81046	75123	67578	0	75513	68352	66380	67926	67833	73619
Space	73722	74667	76263	73752	83175	80132	73510	67045	64808	88997	80881	74824	67389	75513	0	68623	66547	67925	68004	73828
Autos	64869	68272	67034	65523	75352	72281	65265	57222	55249	79468	71974	66355	57696	68352	68623	0	55702	57310	58241	64315
Motorcycles	62607	66092	65034	63537	73272	70385	63305	55552	53331	77562	69850	64323	56068	66380	66547	55702	0	56128	56139	62271
For Sale	64675	68040	66730	65669	75194	72357	63573	55306	53341	77856	70554	65791	56334	67926	67925	57310	56128	0	57783	63427
Baseball	63842	67395	66107	64964	74345	71490	64724	57025	54864	79009	71373	65576	57899	67833	68004	58241	56139	57783	0	61872
Hockey	69760	73563	72111	70996	80163	77388	70458	62783	60878	84967	77265	71736	63987	73619	73828	64315	62271	63427	61872	0

Table 4.4: Distance between original input using UGU metric. UGU values are non-normalized – a fact that will play an important part in its use in the classification task.

We will look at the maximum common subgraph data in more detail to provide some examples. Each row contains an entry which has been bolded – this represents the closest graph in that row as determined by maximum common subgraph. We see that along the diagonal the distances are 0 because two exactly matching graphs should be zero distance. For the Atheism newsgroup, we see that Religion is the closest graph with respect to this particular distance metric. This makes intuitive sense as the topics from such newsgroups would seem to have a large intersection. Other obvious strongly related groups include hockey and baseball, motorcycles and automobiles, and IBM hardware and Macintosh hardware. We also see that the two subcategories of Politics (Guns and Mideast) are most closely related to the Miscellaneous Politics group. Several other interesting relationships exist within the newsgroups, but these experiments are meant to see how well different samples predict these distances rather than exploring the meaning of the distances themselves.

We see in Table 4.3 that the range of the distances provided by the WGU metric are, as expected, more tightly compacted closer to 1 as the ratio is built upon penalizes more than the maximum common subgraph measure as the size of the subgraph becomes smaller and smaller. We still see, though, distances in the same order as that of the maximum common subgraph seen in Table 4.2. The final measure based on the maximum common subgraph (Table 4.4) is not normalized between 0 and 1, but its measures again reflect the previous two. All three metrics based on the maximum common subgraph provide reasonable distance measures between the input graphs.

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	-	**0.8185**	0.9523	0.9600	0.9652	0.9569	0.9744	0.9740	0.9723	0.9799	0.9782	0.9654	0.9709	0.9678	0.9688	0.9702	0.9714	0.9791	0.9701	0.9740
Religion	**0.8185**	-	0.9481	0.9179	0.9566	0.8535	0.9738	0.9727	0.9713	0.9783	0.9778	0.9641	0.9707	0.9576	0.9502	0.9689	0.9708	0.9771	0.9700	0.9755
Christian	0.9523	**0.9481**	-	0.9648	0.9666	0.9607	0.9744	0.9742	0.9730	0.9807	0.9787	0.9689	0.9724	0.9682	0.9721	0.9708	0.9742	0.9785	0.9714	0.9757
Guns	0.9600	0.9179	0.9648	-	0.9474	**0.8926**	0.9709	0.9707	0.9684	0.9776	0.9755	0.9487	0.9682	0.9551	0.9572	0.9621	0.9654	0.9738	0.9659	0.9709
Mideast	0.9652	0.9566	0.9666	0.9474	-	**0.9060**	0.9770	0.9784	0.9761	0.9829	0.9811	0.9681	0.9769	0.9642	0.9650	0.9734	0.9760	0.9813	0.9735	0.9758
Misc Politics	0.9569	**0.8535**	0.9607	0.8926	0.9060	-	0.9721	0.9713	0.9690	0.9776	0.9766	0.9570	0.9692	0.9553	0.9567	0.9642	0.9681	0.9744	0.9659	0.9695
Graphics	0.9744	0.9738	0.9744	0.9709	0.9770	0.9721	-	0.9516	0.9539	0.9610	**0.9479**	0.9640	0.9592	0.9638	0.9620	0.9675	0.9713	0.9632	0.9715	0.9737
IBM Hardware	0.9740	0.9727	0.9742	0.9707	0.9784	0.9713	0.9516	-	**0.9341**	0.9541	0.9612	0.9661	0.9482	0.9672	0.9685	0.9603	0.9673	0.9531	0.9683	0.9712
Mac Hardware	0.9723	0.9713	0.9730	0.9684	0.9761	0.9690	0.9539	**0.9341**	-	0.9646	0.9626	0.9654	0.9498	0.9665	0.9662	0.9598	0.9645	0.9525	0.9662	0.9717
Windows Misc	0.9799	0.9783	0.9807	0.9776	0.9829	0.9776	0.9610	**0.9541**	0.9646	-	0.9689	0.9744	0.9701	0.9756	0.9765	0.9737	0.9772	0.9709	0.9775	0.9803
Windows X	0.9782	0.9778	0.9787	0.9755	0.9811	0.9766	**0.9479**	0.9612	0.9626	0.9689	-	0.9684	0.9663	0.9722	0.9716	0.9725	0.9748	0.9709	0.9758	0.9785
Cryptography	0.9654	0.9641	0.9689	**0.9487**	0.9681	0.9570	0.9640	0.9661	0.9654	0.9744	0.9684	-	0.9629	0.9650	0.9633	0.9671	0.9702	0.9727	0.9692	0.9748
Electronics	0.9709	0.9707	0.9724	0.9682	0.9769	0.9692	0.9592	**0.9482**	0.9498	0.9701	0.9663	0.9629	-	0.9644	0.9634	0.9557	0.9629	0.9541	0.9676	0.9733
Medicine	0.9678	0.9576	0.9682	0.9551	0.9642	0.9553	0.9638	0.9672	0.9665	0.9756	0.9722	0.9650	0.9644	-	**0.9521**	0.9651	0.9685	0.9716	0.9691	0.9718
Space	0.9688	**0.9502**	0.9721	0.9572	0.9650	0.9567	0.9620	0.9685	0.9662	0.9765	0.9716	0.9633	0.9634	0.9521	-	0.9678	0.9706	0.9721	0.9712	0.9739
Autos	0.9702	0.9689	0.9708	0.9621	0.9734	0.9642	0.9675	0.9603	0.9598	0.9737	0.9725	0.9671	0.9557	0.9651	0.9678	-	**0.9522**	0.9572	0.9641	0.9701
Motorcycles	0.9714	0.9708	0.9742	0.9654	0.9760	0.9681	0.9713	0.9673	0.9645	0.9772	0.9748	0.9702	0.9629	0.9685	0.9706	**0.9522**	-	0.9687	0.9668	0.9732
For Sale	0.9791	0.9771	0.9785	0.9738	0.9813	0.9744	0.9632	0.9531	**0.9525**	0.9709	0.9709	0.9727	0.9541	0.9716	0.9721	0.9572	0.9687	-	0.9713	0.9729
Baseball	0.9701	0.9700	0.9714	0.9659	0.9735	0.9659	0.9715	0.9683	0.9662	0.9775	0.9758	0.9692	0.9676	0.9691	0.9712	0.9641	0.9668	0.9713	-	**0.9570**
Hockey	0.9740	0.9755	0.9757	0.9709	0.9758	0.9695	0.9737	0.9712	0.9717	0.9803	0.9785	0.9748	0.9733	0.9718	0.9739	0.9701	0.9732	0.9729	**0.9570**	-

Table 4.5: Distance between original input using Minimum Common Supergraph.

In table 4.5, we see the distances on the original inputs using the minimum common supergraph metric. This technique is based on the smallest possible graph that contains both input graphs (MCS) as well as the largest possible subgraph contained in both input graphs (mcs), and it is defined in formula 4.4. Similar to WGU, MCS penalizes more when the graphs have less and less in common. We see that the graphs measured to be most similar to each other are the same as with the previous three measures, but let us look at a couple of examples in more detail. For the Medicine newsgroup, the maximum common subgraph distances have the top four closest newsgroups as Space, Guns, Miscellaneous Politics, and Religion in that order. The second through fourth closest are hardly separated as well – 0.9250, 0.9267, and 0.9269 for the maximum common subgraph metric. For the minimum common subgraph, space is still the closest newsgroup, and the second through fourth closest are also the same – 0.9551, 0.9553, and 0.9576 for Guns, Miscellaneous Politics, and Religion, respectively.

Another example is the For Sale newsgroup. While the closest group for the maximum common subgraph measures is IBM Hardware, the closest of the minimum common supergraph is Mac Hardware. The distances are extremely close for both measures, and the top five closest are the same as well (IBM Hardware, Mac Hardware, Electronics, Autos, and Motorcycles). These are exactly the situations where we might expect sampling techniques to be tested with respect to their ability to keep accurate estimates of the distances between the original graphs, and we will see later that the sampling does indeed retain the general sense of distance well enough to be used very accurately in tasks such as classification and clustering.

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	-	**0.6087**	0.8712	0.8558	0.8647	0.8442	0.8666	0.8654	0.8672	0.8660	0.8747	0.8607	0.8668	0.8629	0.8656	0.8667	0.8728	0.8784	0.8696	0.8731
Religion	**0.6087**	-	0.8693	0.8017	0.8602	0.6881	0.8738	0.8697	0.8723	0.8705	0.8768	0.8647	0.8687	0.8588	0.8575	0.8722	0.8743	0.8816	0.8741	0.8770
Christian	0.8712	**0.8693**	-	0.8912	0.8940	0.8881	0.8928	0.8964	0.8984	0.8930	0.8956	0.8962	0.8869	0.8847	0.8914	0.8957	0.9012	0.9020	0.9006	0.8992
Guns	0.8558	0.8017	0.8912	-	0.8335	**0.7303**	0.8623	0.8651	0.8628	0.8614	0.8689	0.8382	0.8576	0.8354	0.8445	0.8509	0.8555	0.8669	0.8606	0.8649
Mideast	0.8647	0.8602	0.8940	0.8335	-	**0.7828**	0.9089	0.9187	0.9100	0.9139	0.9139	0.9101	0.9118	0.8961	0.8973	0.9125	0.9161	0.9187	0.9135	0.9145
Misc Politics	0.8442	**0.6881**	0.8881	0.7303	0.7828	-	0.8827	0.8840	0.8806	0.8834	0.8895	0.8610	0.8806	0.8552	0.8630	0.8791	0.8848	0.8910	0.8787	0.8818
Graphics	0.8666	0.8738	0.8928	0.8623	0.9089	0.8827	-	0.8658	0.8735	0.8541	**0.8467**	0.8829	0.8743	0.8769	0.8796	0.8870	0.8956	0.8795	0.8933	0.8936
IBM Hardware	0.8654	0.8697	0.8964	0.8651	0.9187	0.8840	0.8658	-	0.8011	**0.7897**	0.8242	0.8470	0.8084	0.8400	0.8462	0.8376	0.8527	0.8168	0.8486	0.8477
Mac Hardware	0.8672	0.8723	0.8984	0.8628	0.9100	0.8806	0.8735	**0.8011**	-	0.8187	0.8229	0.8370	0.8060	0.8304	0.8230	0.8266	0.8394	0.8097	0.8353	0.8419
Windows Misc	0.8660	0.8705	0.8930	0.8614	0.9139	0.8834	0.8541	**0.7897**	0.8187	-	0.8546	0.8850	0.8732	0.8803	0.8849	0.8799	0.8883	0.8750	0.8864	0.8887
Windows X	0.8747	0.8768	0.8956	0.8689	0.9139	0.8895	0.8467	0.8242	**0.8229**	0.8546	-	0.8849	0.8865	0.8948	0.8922	0.8970	0.8997	0.8915	0.9048	0.9066
Cryptography	0.8607	0.8647	0.8962	0.8382	0.9101	0.8610	0.8829	0.8470	**0.8370**	0.8850	0.8849	-	0.8737	0.8761	0.8750	0.8809	0.8872	0.8866	0.8897	0.8910
Electronics	0.8668	0.8687	0.8869	0.8576	0.9118	0.8806	0.8743	0.8084	**0.8060**	0.8732	0.8865	0.8737	-	0.8272	0.8305	0.8230	0.8416	0.8178	0.8448	0.8469
Medicine	0.8629	0.8588	0.8847	0.8354	0.8961	0.8552	0.8769	0.8400	0.8304	0.8803	0.8948	0.8761	**0.8272**	-	0.8663	0.8781	0.8848	0.8827	0.8829	0.8860
Space	0.8656	0.8575	0.8914	0.8445	0.8973	0.8630	0.8796	0.8462	**0.8230**	0.8849	0.8922	0.8750	0.8305	0.8663	-	0.8817	0.8876	0.8890	0.8867	0.8933
Autos	0.8667	0.8722	0.8957	0.8509	0.9125	0.8791	0.8870	0.8376	0.8266	0.8799	0.8970	0.8809	0.8230	0.8781	0.8817	-	**0.8193**	0.8266	0.8297	0.8420
Motorcycles	0.8728	0.8743	0.9012	0.8555	0.9161	0.8848	0.8956	0.8527	0.8394	0.8883	0.8997	0.8872	0.8416	0.8848	0.8876	**0.8193**	-	0.8459	0.8391	0.8500
For Sale	0.8784	0.8816	0.9020	0.8669	0.9187	0.8910	0.8795	0.8168	**0.8097**	0.8750	0.8915	0.8866	0.8178	0.8827	0.8890	0.8266	0.8459	-	0.8637	0.8668
Baseball	0.8696	0.8741	0.9006	0.8606	0.9135	0.8787	0.8933	0.8486	0.8353	0.8864	0.9048	0.8897	0.8448	0.8829	0.8867	0.8297	0.8391	0.8637	-	**0.8190**
Hockey	0.8731	0.8770	0.8992	0.8649	0.9145	0.8818	0.8936	0.8477	0.8419	0.8887	0.9066	0.8910	0.8469	0.8860	0.8933	0.8420	0.8500	0.8668	**0.8190**	-

Table 4.6: Distance between original input using Sample Intersection metric. The SID distance measure appears to do an excellent job of approximating the maximum common subgraph metric.

The final distance measure is the Sample Intersection distance (SID) shown in table 4.6. This metric is introduced here as a way to estimate quickly the similarity between two graphs by examining the edge list randomly, and it can be performed more efficiently than the previously discussed measures because it does not require an exhaustive search of all edges and nodes. In addition, the number of random edges checked can be tuned to affect the tradeoff between accuracy and efficiency. For these experiments (as well as for the sample experiments discussed later), the number of edges randomly chosen to check for presence in both graphs was set at 30% of the number of edges of the larger of the two graphs.

For the most part, the distances using the SID metric are very similar overall to the previous four. One interesting recurring difference is that the Mac Hardware newsgroup graph appears as the closest graph to several other graphs to which it's close but not *the* closest when using the other measures. The general groupings are very similar, though, and again the emphasis should be on how well the samples recreate the relative distances for each metric as opposed to the relative qualities of the metrics themselves.

Hybrid Node-Edge Sample Distances

Let us first examine the ability of the node-edge sampling technique to retain the graph similarity properties of the original inputs. The distances between the proxy graphs are ordered remarkably closely to the original inputs. The size of the graph has been significantly reduced, but the relative distances between the graphs has hardly changed at all. We will see the magnitude of the changes in the relative distance rankings for both sampling techniques in Figures 4.2, 4.3, and 4.4, but first we present the raw distances between node-edge samples of

15, 30, and 50% (Tables 4.7, 4.8, and 4.9) for the maximum common subgraph distance. The other metrics for all sample sizes may be found in Appendix A.

The proxies track the distances between respective input graphs faithfully. As expected, larger proxies are even more accurate in maintaining relative distance. Across all sample sizes, all metrics perform quite well. We will see that this translates to very good performance in classification in section 4.6.

Random Walk Sample Distances

Now let us take a look at the random walk sample's ability to retain the graph similarity properties of the original inputs. Tables 4.10, 4.11, and 4.12 provide distance data for the 15, 30, and 50% samples respectively. We see the distances between the sample data using the maximum common subgraph distance measure, but the other detailed distance data for the WGU, UGUG, minimum common subgraph, and sample intersection distance may be found in Appendix A. .

Just as with the node-edge method, the random walk does a very good job of preserving relative ordering between the graphs. While the absolute numbers change a little, the relative distances are very similar due to the ability of SLIP to maintain problem specific features. We will see in histograms that this implementation of the random walk has a bit more variation than the node-edge technique, but there are several parameters that can be adjusted to improve the random walk's effectiveness. Such parameters include the probability of returning to the same start node as well as the maximum path length.

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.7767	0.8489	0.8657	0.8871	0.8707	0.9083	0.9135	0.9142	0.9293	0.9195	0.8816	0.9026	0.8912	0.8988	0.9008	0.9108	0.9323	0.8990	0.9081
Religion	0.7767	0.0000	0.8411	0.8366	0.8730	0.8067	0.9115	0.9136	0.9210	0.9241	0.9201	0.8833	0.9089	0.8828	0.8808	0.9067	0.9130	0.9324	0.9067	0.9159
Christian	0.8489	0.8411	0.0000	0.8706	0.8879	0.8695	0.9075	0.9164	0.9225	0.9306	0.9174	0.8867	0.9095	0.8857	0.8980	0.9121	0.9206	0.9346	0.9085	0.9170
Guns	0.8657	0.8366	0.8706	0.0000	0.8671	0.8140	0.8979	0.9085	0.9135	0.9230	0.9145	0.8552	0.8995	0.8746	0.8750	0.8908	0.9005	0.9240	0.8951	0.9005
Mideast	0.8871	0.8730	0.8879	0.8671	0.0000	0.8118	0.9272	0.9320	0.9310	0.9309	0.9258	0.8902	0.9247	0.8975	0.8934	0.9192	0.9268	0.9442	0.9205	0.9202
Misc Politics	0.8707	0.8067	0.8695	0.8140	0.8118	0.0000	0.9069	0.9132	0.9147	0.9142	0.9113	0.8654	0.9057	0.8678	0.8741	0.8999	0.9116	0.9276	0.9022	0.9050
Graphics	0.9083	0.9115	0.9075	0.8979	0.9272	0.9069	0.0000	0.8757	0.8839	0.8977	0.8648	0.8792	0.8837	0.8935	0.8850	0.9042	0.9179	0.9018	0.9118	0.9122
IBM Hardware	0.9135	0.9136	0.9164	0.9085	0.9320	0.9132	0.8757	0.0000	0.8319	0.9008	0.8967	0.8971	0.8578	0.9066	0.9067	0.8853	0.8943	0.8727	0.8922	0.9117
Mac Hardware	0.9142	0.9210	0.9225	0.9135	0.9310	0.9147	0.8839	0.8319	0.0000	0.9134	0.9010	0.9033	0.8623	0.9095	0.9066	0.8861	0.8958	0.8773	0.8948	0.9179
Windows Misc	0.9293	0.9241	0.9306	0.9230	0.9309	0.9142	0.8977	0.9008	0.9134	0.0000	0.8962	0.9124	0.9122	0.9169	0.9176	0.9259	0.9319	0.9227	0.9314	0.9372
Windows X	0.9195	0.9201	0.9174	0.9145	0.9258	0.9113	0.8648	0.8967	0.9010	0.8962	0.0000	0.8947	0.9012	0.9038	0.8952	0.9180	0.9255	0.9158	0.9248	0.9255
Cryptography	0.8816	0.8833	0.8867	0.8552	0.8902	0.8654	0.8792	0.8971	0.9033	0.9124	0.8947	0.0000	0.8892	0.8804	0.8784	0.8998	0.9076	0.9189	0.9000	0.9108
Electronics	0.9026	0.9089	0.9095	0.8995	0.9247	0.9057	0.8837	0.8578	0.8623	0.9122	0.9012	0.8892	0.0000	0.8976	0.8936	0.8745	0.8852	0.8763	0.8884	0.9086
Medicine	0.8912	0.8828	0.8857	0.8746	0.8975	0.8678	0.8935	0.9066	0.9095	0.9169	0.9038	0.8804	0.8976	0.0000	0.8726	0.9031	0.9120	0.9226	0.9081	0.9073
Space	0.8988	0.8808	0.8980	0.8750	0.8934	0.8741	0.8850	0.9067	0.9066	0.9176	0.8952	0.8784	0.8936	0.8726	0.0000	0.9030	0.9150	0.9173	0.9100	0.9105
Autos	0.9008	0.9067	0.9121	0.8908	0.9192	0.8999	0.9042	0.8853	0.8861	0.9259	0.9180	0.8998	0.8745	0.9031	0.9030	0.0000	0.8690	0.8916	0.8827	0.9067
Motorcycles	0.9108	0.9130	0.9206	0.9005	0.9268	0.9116	0.9179	0.8943	0.8958	0.9319	0.9255	0.9076	0.8852	0.9120	0.9150	0.8690	0.0000	0.9052	0.8917	0.9198
For Sale	0.9323	0.9324	0.9346	0.9240	0.9442	0.9276	0.9018	0.8727	0.8773	0.9227	0.9158	0.9189	0.8763	0.9226	0.9173	0.8916	0.9052	0.0000	0.9074	0.9181
Baseball	0.8990	0.9067	0.9085	0.8951	0.9205	0.9022	0.9118	0.8922	0.8948	0.9314	0.9248	0.9000	0.8884	0.9081	0.9100	0.8827	0.8917	0.9074	0.0000	0.8687
Hockey	0.9081	0.9159	0.9170	0.9005	0.9202	0.9050	0.9122	0.9117	0.9179	0.9372	0.9255	0.9108	0.9086	0.9073	0.9105	0.9067	0.9198	0.9181	0.8687	0.0000

Table 4.7: mcs Distance between Node-Edge Samples (15%). Although the absolute values are different than those for the mcs metric for the original graphs, the relative values are similar.

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.7497	0.8539	0.8778	0.8992	0.8808	0.9183	0.9208	0.9199	0.9386	0.9307	0.8926	0.9126	0.9055	0.9046	0.9104	0.9200	0.9390	0.9101	0.9176
Religion	0.7497	0.0000	0.8479	0.8294	0.8824	0.7844	0.9184	0.9210	0.9239	0.9320	0.9282	0.8911	0.9153	0.8877	0.8810	0.9140	0.9191	0.9391	0.9133	0.9256
Christian	0.8539	0.8479	0.0000	0.8851	0.8973	0.8819	0.9195	0.9239	0.9259	0.9395	0.9307	0.8988	0.9176	0.8986	0.9069	0.9174	0.9242	0.9390	0.9160	0.9248
Guns	0.8778	0.8294	0.8851	0.0000	0.8692	0.8023	0.9095	0.9136	0.9143	0.9342	0.9252	0.8639	0.9080	0.8795	0.8810	0.8985	0.9073	0.9299	0.9041	0.9113
Mideast	0.8992	0.8824	0.8973	0.8692	0.0000	0.8133	0.9341	0.9432	0.9376	0.9425	0.9381	0.9060	0.9355	0.9041	0.9044	0.9291	0.9355	0.9512	0.9305	0.9308
Misc Politics	0.8808	0.7844	0.8819	0.8023	0.8133	0.0000	0.9208	0.9236	0.9186	0.9280	0.9238	0.8818	0.9157	0.8825	0.8805	0.9081	0.9175	0.9335	0.9115	0.9151
Graphics	0.9183	0.9184	0.9195	0.9095	0.9341	0.9208	0.0000	0.8856	0.8884	0.9031	0.8761	0.8916	0.8959	0.9065	0.8928	0.9137	0.9208	0.9056	0.9210	0.9219
IBM Hardware	0.9208	0.9210	0.9239	0.9136	0.9432	0.9236	0.8856	0.0000	0.8405	0.9019	0.9038	0.9095	0.8644	0.9157	0.9149	0.8908	0.8985	0.8738	0.9010	0.9213
Mac Hardware	0.9199	0.9239	0.9259	0.9143	0.9376	0.9186	0.8884	0.8405	0.0000	0.9173	0.9083	0.9100	0.8695	0.9129	0.9104	0.8921	0.8962	0.8773	0.8999	0.9240
Windows Misc	0.9386	0.9320	0.9395	0.9342	0.9425	0.9280	0.9031	0.9019	0.9173	0.0000	0.9115	0.9250	0.9240	0.9263	0.9275	0.9329	0.9404	0.9296	0.9393	0.9438
Windows X	0.9307	0.9282	0.9307	0.9252	0.9381	0.9238	0.8761	0.9038	0.9083	0.9115	0.0000	0.9075	0.9137	0.9158	0.9121	0.9263	0.9324	0.9264	0.9331	0.9368
Cryptography	0.8926	0.8911	0.8988	0.8639	0.9060	0.8818	0.8916	0.9095	0.9100	0.9250	0.9075	0.0000	0.8999	0.8955	0.8863	0.9086	0.9185	0.9256	0.9169	0.9246
Electronics	0.9126	0.9153	0.9176	0.9080	0.9355	0.9157	0.8959	0.8644	0.8695	0.9240	0.9137	0.8999	0.0000	0.9065	0.9018	0.8766	0.8909	0.8842	0.8987	0.9216
Medicine	0.9055	0.8877	0.8986	0.8795	0.9041	0.8825	0.9065	0.9157	0.9129	0.9263	0.9158	0.8955	0.9065	0.0000	0.8776	0.9082	0.9170	0.9267	0.9142	0.9211
Space	0.9046	0.8810	0.9069	0.8810	0.9044	0.8805	0.8928	0.9149	0.9104	0.9275	0.9121	0.8863	0.9018	0.8776	0.0000	0.9080	0.9204	0.9228	0.9176	0.9217
Autos	0.9104	0.9140	0.9174	0.8985	0.9291	0.9081	0.9137	0.8908	0.8921	0.9329	0.9263	0.9086	0.8766	0.9082	0.9080	0.0000	0.8681	0.8947	0.8929	0.9134
Motorcycles	0.9200	0.9191	0.9242	0.9073	0.9355	0.9175	0.9208	0.8985	0.8962	0.9404	0.9324	0.9185	0.8909	0.9170	0.9204	0.8681	0.0000	0.9123	0.8980	0.9239
For Sale	0.9390	0.9391	0.9390	0.9299	0.9512	0.9335	0.9056	0.8738	0.8773	0.9296	0.9264	0.9256	0.8842	0.9267	0.9228	0.8947	0.9123	0.0000	0.9154	0.9243
Baseball	0.9101	0.9133	0.9160	0.9041	0.9305	0.9115	0.9210	0.9010	0.8999	0.9393	0.9331	0.9169	0.8987	0.9142	0.9176	0.8929	0.8980	0.9154	0.0000	0.8830
Hockey	0.9176	0.9256	0.9248	0.9113	0.9308	0.9151	0.9219	0.9213	0.9240	0.9438	0.9368	0.9246	0.9216	0.9211	0.9217	0.9134	0.9239	0.9243	0.8830	0.0000

Table 4.8: mcs Distance between Node-Edge Samples (30%). For this larger sample size, the relative mcs distance ordering between each graph is approximated very well by the proxy.

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.7358	0.8806	0.8964	0.9178	0.8983	0.9317	0.9349	0.9364	0.9521	0.9445	0.9102	0.9273	0.9186	0.9205	0.9239	0.9312	0.9480	0.9271	0.9319
Religion	0.7358	0.0000	0.8732	0.8416	0.9007	0.7766	0.9323	0.9359	0.9357	0.9479	0.9413	0.9081	0.9309	0.8998	0.8941	0.9254	0.9319	0.9479	0.9299	0.9380
Christian	0.8806	0.8732	0.0000	0.9053	0.9184	0.9028	0.9318	0.9372	0.9382	0.9527	0.9427	0.9158	0.9318	0.9150	0.9237	0.9291	0.9373	0.9488	0.9304	0.9367
Guns	0.8964	0.8416	0.9053	0.0000	0.8830	0.8044	0.9228	0.9272	0.9282	0.9466	0.9381	0.8794	0.9221	0.8921	0.8966	0.9101	0.9206	0.9394	0.9183	0.9236
Mideast	0.9178	0.9007	0.9184	0.8830	0.0000	0.8189	0.9440	0.9523	0.9491	0.9547	0.9507	0.9230	0.9481	0.9168	0.9193	0.9408	0.9475	0.9585	0.9428	0.9423
Misc Politics	0.8983	0.7766	0.9028	0.8044	0.8189	0.0000	0.9330	0.9373	0.9335	0.9427	0.9374	0.8992	0.9298	0.8953	0.8976	0.9225	0.9322	0.9444	0.9260	0.9281
Graphics	0.9317	0.9323	0.9318	0.9228	0.9440	0.9330	0.0000	0.8967	0.9030	0.9175	0.8929	0.9086	0.9127	0.9163	0.9115	0.9235	0.9324	0.9180	0.9328	0.9321
IBM Hardware	0.9349	0.9359	0.9372	0.9272	0.9523	0.9373	0.8967	0.0000	0.8596	0.9153	0.9183	0.9225	0.8820	0.9253	0.9276	0.9043	0.9164	0.8891	0.9164	0.9320
Mac Hardware	0.9364	0.9357	0.9382	0.9282	0.9491	0.9335	0.9030	0.8596	0.0000	0.9321	0.9235	0.9245	0.8864	0.9281	0.9263	0.9086	0.9121	0.8924	0.9149	0.9357
Windows Misc	0.9521	0.9479	0.9527	0.9466	0.9547	0.9427	0.9175	0.9153	0.9321	0.0000	0.9253	0.9395	0.9390	0.9423	0.9423	0.9442	0.9515	0.9410	0.9515	0.9554
Windows X	0.9445	0.9413	0.9427	0.9381	0.9507	0.9374	0.8929	0.9183	0.9235	0.9253	0.0000	0.9224	0.9278	0.9292	0.9284	0.9386	0.9435	0.9370	0.9447	0.9477
Cryptography	0.9102	0.9081	0.9158	0.8794	0.9230	0.8992	0.9086	0.9225	0.9245	0.9395	0.9224	0.0000	0.9150	0.9112	0.9061	0.9222	0.9307	0.9372	0.9278	0.9358
Electronics	0.9273	0.9309	0.9318	0.9221	0.9481	0.9298	0.9127	0.8820	0.8864	0.9390	0.9278	0.9150	0.0000	0.9219	0.9168	0.8934	0.9060	0.8960	0.9149	0.9346
Medicine	0.9186	0.8998	0.9150	0.8921	0.9168	0.8953	0.9163	0.9253	0.9281	0.9423	0.9292	0.9112	0.9219	0.0000	0.8924	0.9199	0.9306	0.9372	0.9290	0.9309
Space	0.9205	0.8941	0.9237	0.8966	0.9193	0.8976	0.9115	0.9276	0.9263	0.9423	0.9284	0.9061	0.9168	0.8924	0.0000	0.9248	0.9332	0.9368	0.9326	0.9324
Autos	0.9239	0.9254	0.9291	0.9101	0.9408	0.9225	0.9235	0.9043	0.9086	0.9442	0.9386	0.9222	0.8934	0.9199	0.9248	0.0000	0.8850	0.9057	0.9086	0.9283
Motorcycles	0.9312	0.9319	0.9373	0.9206	0.9475	0.9322	0.9324	0.9164	0.9121	0.9515	0.9435	0.9307	0.9060	0.9306	0.9332	0.8850	0.0000	0.9237	0.9149	0.9359
For Sale	0.9480	0.9479	0.9488	0.9394	0.9585	0.9444	0.9180	0.8891	0.8924	0.9410	0.9370	0.9372	0.8960	0.9372	0.9368	0.9057	0.9237	0.0000	0.9281	0.9359
Baseball	0.9271	0.9299	0.9304	0.9183	0.9428	0.9260	0.9328	0.9164	0.9149	0.9515	0.9447	0.9278	0.9149	0.9290	0.9326	0.9086	0.9149	0.9281	0.0000	0.8993
Hockey	0.9319	0.9380	0.9367	0.9236	0.9423	0.9281	0.9321	0.9320	0.9357	0.9554	0.9477	0.9358	0.9346	0.9309	0.9324	0.9283	0.9359	0.9359	0.8993	0.0000

Table 4.9: mcs Distance between Node-Edge Samples (50%). For the largest sample size, we see that the absolute distances are a little closer but the relative rankings of the distances between graphs is a highly accurate approximation.

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.8596	0.9473	0.9507	0.9614	0.9603	0.9642	0.9679	0.9668	0.9668	0.9728	0.9528	0.9597	0.9594	0.9634	0.9584	0.9602	0.9736	0.9625	0.9634
Religion	0.8596	0.0000	0.9473	0.9256	0.9565	0.9029	0.9621	0.9653	0.9616	0.9589	0.9717	0.9540	0.9554	0.9505	0.9558	0.9557	0.9589	0.9704	0.9646	0.9632
Christian	0.9473	0.9473	0.0000	0.9622	0.9664	0.9610	0.9682	0.9780	0.9720	0.9722	0.9744	0.9620	0.9621	0.9625	0.9684	0.9653	0.9665	0.9753	0.9705	0.9703
Guns	0.9507	0.9256	0.9622	0.0000	0.9552	0.9197	0.9601	0.9654	0.9660	0.9635	0.9716	0.9458	0.9575	0.9506	0.9573	0.9523	0.9542	0.9696	0.9622	0.9633
Mideast	0.9614	0.9565	0.9664	0.9552	0.0000	0.9307	0.9716	0.9764	0.9731	0.9681	0.9754	0.9612	0.9681	0.9657	0.9671	0.9667	0.9671	0.9792	0.9699	0.9694
Misc Politics	0.9603	0.9029	0.9610	0.9197	0.9307	0.0000	0.9679	0.9724	0.9682	0.9642	0.9727	0.9582	0.9638	0.9543	0.9606	0.9624	0.9667	0.9760	0.9667	0.9697
Graphics	0.9642	0.9621	0.9682	0.9601	0.9716	0.9679	0.0000	0.9596	0.9586	0.9492	0.9532	0.9573	0.9522	0.9591	0.9609	0.9611	0.9587	0.9633	0.9690	0.9668
IBM Hardware	0.9679	0.9653	0.9780	0.9654	0.9764	0.9724	0.9596	0.0000	0.9442	0.9525	0.9680	0.9643	0.9451	0.9661	0.9698	0.9544	0.9525	0.9571	0.9632	0.9703
Mac Hardware	0.9668	0.9616	0.9720	0.9660	0.9731	0.9682	0.9586	0.9442	0.0000	0.9598	0.9668	0.9638	0.9419	0.9616	0.9648	0.9543	0.9476	0.9591	0.9593	0.9697
Windows Misc	0.9668	0.9589	0.9722	0.9635	0.9681	0.9642	0.9492	0.9525	0.9598	0.0000	0.9580	0.9614	0.9560	0.9596	0.9650	0.9642	0.9617	0.9677	0.9684	0.9688
Windows X	0.9728	0.9717	0.9744	0.9716	0.9754	0.9727	0.9532	0.9680	0.9668	0.9580	0.0000	0.9666	0.9644	0.9671	0.9683	0.9701	0.9714	0.9751	0.9741	0.9770
Cryptography	0.9528	0.9540	0.9620	0.9458	0.9612	0.9582	0.9573	0.9643	0.9638	0.9614	0.9666	0.0000	0.9559	0.9585	0.9568	0.9565	0.9595	0.9699	0.9668	0.9680
Electronics	0.9597	0.9554	0.9621	0.9575	0.9681	0.9638	0.9522	0.9451	0.9419	0.9560	0.9644	0.9559	0.0000	0.9528	0.9561	0.9431	0.9402	0.9496	0.9535	0.9624
Medicine	0.9594	0.9505	0.9625	0.9506	0.9657	0.9543	0.9591	0.9661	0.9616	0.9596	0.9671	0.9585	0.9528	0.0000	0.9551	0.9560	0.9578	0.9671	0.9655	0.9640
Space	0.9634	0.9558	0.9684	0.9573	0.9671	0.9606	0.9609	0.9698	0.9648	0.9650	0.9683	0.9568	0.9561	0.9551	0.0000	0.9611	0.9654	0.9729	0.9677	0.9668
Autos	0.9584	0.9557	0.9653	0.9523	0.9667	0.9624	0.9611	0.9544	0.9543	0.9642	0.9701	0.9565	0.9431	0.9560	0.9611	0.0000	0.9348	0.9580	0.9540	0.9626
Motorcycles	0.9602	0.9589	0.9665	0.9542	0.9671	0.9667	0.9587	0.9525	0.9476	0.9617	0.9714	0.9595	0.9402	0.9578	0.9654	0.9348	0.0000	0.9609	0.9538	0.9606
For Sale	0.9736	0.9704	0.9753	0.9696	0.9792	0.9760	0.9633	0.9571	0.9591	0.9677	0.9751	0.9699	0.9496	0.9671	0.9729	0.9580	0.9609	0.0000	0.9660	0.9689
Baseball	0.9625	0.9646	0.9705	0.9622	0.9699	0.9667	0.9690	0.9632	0.9593	0.9684	0.9741	0.9668	0.9535	0.9655	0.9677	0.9540	0.9538	0.9660	0.0000	0.9555
Hockey	0.9634	0.9632	0.9703	0.9633	0.9694	0.9697	0.9668	0.9703	0.9697	0.9688	0.9770	0.9680	0.9624	0.9640	0.9668	0.9626	0.9606	0.9689	0.9555	0.0000

Table 4.10: mcs Distance between Random Walk Samples (15%). The smallest proxy created using random walk sampling does a good job of approximating the relative ordering of distances between the newsgroups, but we will see that generally the node-edge sampling technique outperforms it in accuracy.

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.8596	0.9473	0.9507	0.9614	0.9603	0.9642	0.9679	0.9668	0.9668	0.9728	0.9528	0.9597	0.9594	0.9634	0.9584	0.9602	0.9736	0.9625	0.9634
Religion	0.8596	0.0000	0.9473	0.9256	0.9565	0.9029	0.9621	0.9653	0.9616	0.9589	0.9717	0.9540	0.9554	0.9505	0.9558	0.9557	0.9589	0.9704	0.9646	0.9632
Christian	0.9473	0.9473	0.0000	0.9622	0.9664	0.9610	0.9682	0.9780	0.9720	0.9722	0.9744	0.9620	0.9621	0.9625	0.9684	0.9653	0.9665	0.9753	0.9705	0.9703
Guns	0.9507	0.9256	0.9622	0.0000	0.9552	0.9197	0.9601	0.9654	0.9660	0.9635	0.9716	0.9458	0.9575	0.9506	0.9573	0.9523	0.9542	0.9696	0.9622	0.9633
Mideast	0.9614	0.9565	0.9664	0.9552	0.0000	0.9307	0.9716	0.9764	0.9731	0.9681	0.9754	0.9612	0.9681	0.9657	0.9671	0.9667	0.9671	0.9792	0.9699	0.9694
Misc Politics	0.9603	0.9029	0.9610	0.9197	0.9307	0.0000	0.9679	0.9724	0.9682	0.9642	0.9727	0.9582	0.9638	0.9543	0.9606	0.9624	0.9667	0.9760	0.9667	0.9697
Graphics	0.9642	0.9621	0.9682	0.9601	0.9716	0.9679	0.0000	0.9596	0.9586	0.9492	0.9532	0.9573	0.9522	0.9591	0.9609	0.9611	0.9587	0.9633	0.9690	0.9668
IBM Hardware	0.9679	0.9653	0.9780	0.9654	0.9764	0.9724	0.9596	0.0000	0.9442	0.9525	0.9680	0.9643	0.9451	0.9661	0.9698	0.9544	0.9525	0.9571	0.9632	0.9703
Mac Hardware	0.9668	0.9616	0.9720	0.9660	0.9731	0.9682	0.9586	0.9442	0.0000	0.9598	0.9668	0.9638	0.9419	0.9616	0.9648	0.9543	0.9476	0.9591	0.9593	0.9697
Windows Misc	0.9668	0.9589	0.9722	0.9635	0.9681	0.9642	0.9492	0.9525	0.9598	0.0000	0.9580	0.9614	0.9560	0.9596	0.9650	0.9642	0.9617	0.9677	0.9684	0.9688
Windows X	0.9728	0.9717	0.9744	0.9716	0.9754	0.9727	0.9532	0.9680	0.9668	0.9580	0.0000	0.9666	0.9644	0.9671	0.9683	0.9701	0.9714	0.9751	0.9741	0.9770
Cryptography	0.9528	0.9540	0.9620	0.9458	0.9612	0.9582	0.9573	0.9643	0.9638	0.9614	0.9666	0.0000	0.9559	0.9585	0.9568	0.9565	0.9595	0.9699	0.9668	0.9680
Electronics	0.9597	0.9554	0.9621	0.9575	0.9681	0.9638	0.9522	0.9451	0.9419	0.9560	0.9644	0.9559	0.0000	0.9528	0.9561	0.9431	0.9402	0.9496	0.9535	0.9624
Medicine	0.9594	0.9505	0.9625	0.9506	0.9657	0.9543	0.9591	0.9661	0.9616	0.9596	0.9671	0.9585	0.9528	0.0000	0.9551	0.9560	0.9578	0.9671	0.9655	0.9640
Space	0.9634	0.9558	0.9684	0.9573	0.9671	0.9606	0.9609	0.9698	0.9648	0.9650	0.9683	0.9568	0.9561	0.9551	0.0000	0.9611	0.9654	0.9729	0.9677	0.9668
Autos	0.9584	0.9557	0.9653	0.9523	0.9667	0.9624	0.9611	0.9544	0.9543	0.9642	0.9701	0.9565	0.9431	0.9560	0.9611	0.0000	0.9348	0.9580	0.9540	0.9626
Motorcycles	0.9602	0.9589	0.9665	0.9542	0.9671	0.9667	0.9587	0.9525	0.9476	0.9617	0.9714	0.9595	0.9402	0.9578	0.9654	0.9348	0.0000	0.9609	0.9538	0.9606
For Sale	0.9736	0.9704	0.9753	0.9696	0.9792	0.9760	0.9633	0.9571	0.9591	0.9677	0.9751	0.9699	0.9496	0.9671	0.9729	0.9580	0.9609	0.0000	0.9660	0.9689
Baseball	0.9625	0.9646	0.9705	0.9622	0.9699	0.9667	0.9690	0.9632	0.9593	0.9684	0.9741	0.9668	0.9535	0.9655	0.9677	0.9540	0.9538	0.9660	0.0000	0.9555
Hockey	0.9634	0.9632	0.9703	0.9633	0.9694	0.9697	0.9668	0.9703	0.9697	0.9688	0.9770	0.9680	0.9624	0.9640	0.9668	0.9626	0.9606	0.9689	0.9555	0.0000

Table 4.11: mcs Distance between Random Walk Samples (30%). As one would expect, the random walk samples become more accurate in their approximations with larger sample sizes.

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.7946	0.9363	0.9448	0.9566	0.9472	0.9593	0.9589	0.9601	0.9642	0.9666	0.9496	0.9529	0.9542	0.9559	0.9565	0.9602	0.9681	0.9557	0.9606
Religion	0.7946	0.0000	0.9305	0.9061	0.9507	0.8593	0.9582	0.9573	0.9584	0.9582	0.9647	0.9447	0.9539	0.9417	0.9465	0.9523	0.9588	0.9665	0.9555	0.9628
Christian	0.9363	0.9305	0.0000	0.9525	0.9598	0.9529	0.9625	0.9628	0.9654	0.9670	0.9697	0.9554	0.9571	0.9542	0.9609	0.9596	0.9648	0.9734	0.9606	0.9661
Guns	0.9448	0.9061	0.9525	0.0000	0.9471	0.8935	0.9582	0.9578	0.9575	0.9609	0.9658	0.9364	0.9509	0.9430	0.9479	0.9487	0.9546	0.9652	0.9522	0.9587
Mideast	0.9566	0.9507	0.9598	0.9471	0.0000	0.9104	0.9703	0.9732	0.9718	0.9677	0.9720	0.9610	0.9687	0.9587	0.9604	0.9666	0.9712	0.9781	0.9664	0.9700
Misc Politics	0.9472	0.8593	0.9529	0.8935	0.9104	0.0000	0.9640	0.9629	0.9625	0.9592	0.9667	0.9449	0.9576	0.9444	0.9469	0.9554	0.9620	0.9699	0.9581	0.9645
Graphics	0.9593	0.9582	0.9625	0.9582	0.9703	0.9640	0.0000	0.9411	0.9461	0.9416	0.9397	0.9535	0.9441	0.9527	0.9511	0.9521	0.9595	0.9559	0.9604	0.9622
IBM Hardware	0.9589	0.9573	0.9628	0.9578	0.9732	0.9629	0.9411	0.0000	0.9176	0.9374	0.9573	0.9556	0.9240	0.9560	0.9579	0.9385	0.9482	0.9384	0.9482	0.9589
Mac Hardware	0.9601	0.9584	0.9654	0.9575	0.9718	0.9625	0.9461	0.9176	0.0000	0.9514	0.9572	0.9549	0.9246	0.9553	0.9557	0.9411	0.9465	0.9398	0.9488	0.9602
Windows Misc	0.9642	0.9582	0.9670	0.9609	0.9677	0.9592	0.9416	0.9374	0.9514	0.0000	0.9487	0.9588	0.9514	0.9569	0.9583	0.9573	0.9645	0.9604	0.9637	0.9675
Windows X	0.9666	0.9647	0.9697	0.9658	0.9720	0.9667	0.9397	0.9573	0.9572	0.9487	0.0000	0.9588	0.9549	0.9604	0.9588	0.9626	0.9662	0.9670	0.9687	0.9694
Cryptography	0.9496	0.9447	0.9554	0.9364	0.9610	0.9449	0.9535	0.9556	0.9549	0.9588	0.9588	0.0000	0.9475	0.9505	0.9484	0.9526	0.9592	0.9644	0.9573	0.9634
Electronics	0.9529	0.9539	0.9571	0.9509	0.9687	0.9576	0.9441	0.9240	0.9246	0.9514	0.9549	0.9475	0.0000	0.9486	0.9479	0.9271	0.9400	0.9380	0.9432	0.9567
Medicine	0.9542	0.9417	0.9542	0.9430	0.9587	0.9444	0.9527	0.9560	0.9553	0.9569	0.9604	0.9505	0.9486	0.0000	0.9416	0.9521	0.9574	0.9619	0.9559	0.9620
Space	0.9559	0.9465	0.9609	0.9479	0.9604	0.9469	0.9511	0.9579	0.9557	0.9583	0.9588	0.9484	0.9479	0.9416	0.0000	0.9556	0.9618	0.9662	0.9609	0.9635
Autos	0.9565	0.9523	0.9596	0.9487	0.9666	0.9554	0.9521	0.9385	0.9411	0.9573	0.9626	0.9526	0.9271	0.9521	0.9556	0.0000	0.9338	0.9424	0.9420	0.9556
Motorcycles	0.9602	0.9588	0.9648	0.9546	0.9712	0.9620	0.9595	0.9482	0.9465	0.9645	0.9662	0.9592	0.9400	0.9574	0.9618	0.9338	0.0000	0.9563	0.9481	0.9622
For Sale	0.9681	0.9665	0.9734	0.9652	0.9781	0.9699	0.9559	0.9384	0.9398	0.9604	0.9670	0.9644	0.9380	0.9619	0.9662	0.9424	0.9563	0.0000	0.9553	0.9634
Baseball	0.9557	0.9555	0.9606	0.9522	0.9664	0.9581	0.9604	0.9482	0.9488	0.9637	0.9687	0.9573	0.9432	0.9559	0.9609	0.9420	0.9481	0.9553	0.0000	0.9429
Hockey	0.9606	0.9628	0.9661	0.9587	0.9700	0.9645	0.9622	0.9589	0.9602	0.9675	0.9694	0.9634	0.9567	0.9620	0.9635	0.9556	0.9622	0.9634	0.9429	0.0000

Table 4.12: mcs Distance between Random Walk Samples (50%). The largest sample size using a random walk does the best job of approximating the mcs distances between the graphs based on the original inputs.

4.5.5 Results

Figures 4.2, 4.3, and 4.4 provide histograms detailing the changes in relative rankings between the proxy and the original for both the node-edge and the random walk samples for all five distance metrics and all three sample rates. If the original input had the atheism and religion newsgroups as closest to each other and the sample had them closest together as well, the change in proximity ranking from input to sample is 19 (20 is always itself). If the sample had another group closer to religion than atheism (say atheism was the second closest), the change in the relative ranking is -1.

These figures provide a good summary view of the preservation of the PSF of graph distance. We see from the results that proxies created from the original graphs do an excellent job of maintaining relative ordering with respect to the distance metric used. As the proxies become smaller, the changes in distance rankings become more pronounced. With the random walk proxy in particular, the inaccuracy of the 15% proxy becomes more pronounced. For the 30 and 50% samples, the variations that do exist are minor (particularly for the node-edge random sampling technique), and they are attributed to the original distances between certain groups being clustered into close groups initially. We might expect then that a sample might flip an ordering here and there, but the same neighborhoods exist in the rankings.

Let us look at the 30% proxy in more detail since it is the one that will be used in the classification task in the next section. For the node-edge sampling, only the SID metric has many pairs whose ranking changes more than five spots, showing that the samples are similarly structured despite being significantly smaller. For the random walk, we see that the tail is thicker which tells us that more pairs of graphs deviate more widely in the ranking change from input to proxy. The distance data as well as the analysis of ranking changes show that

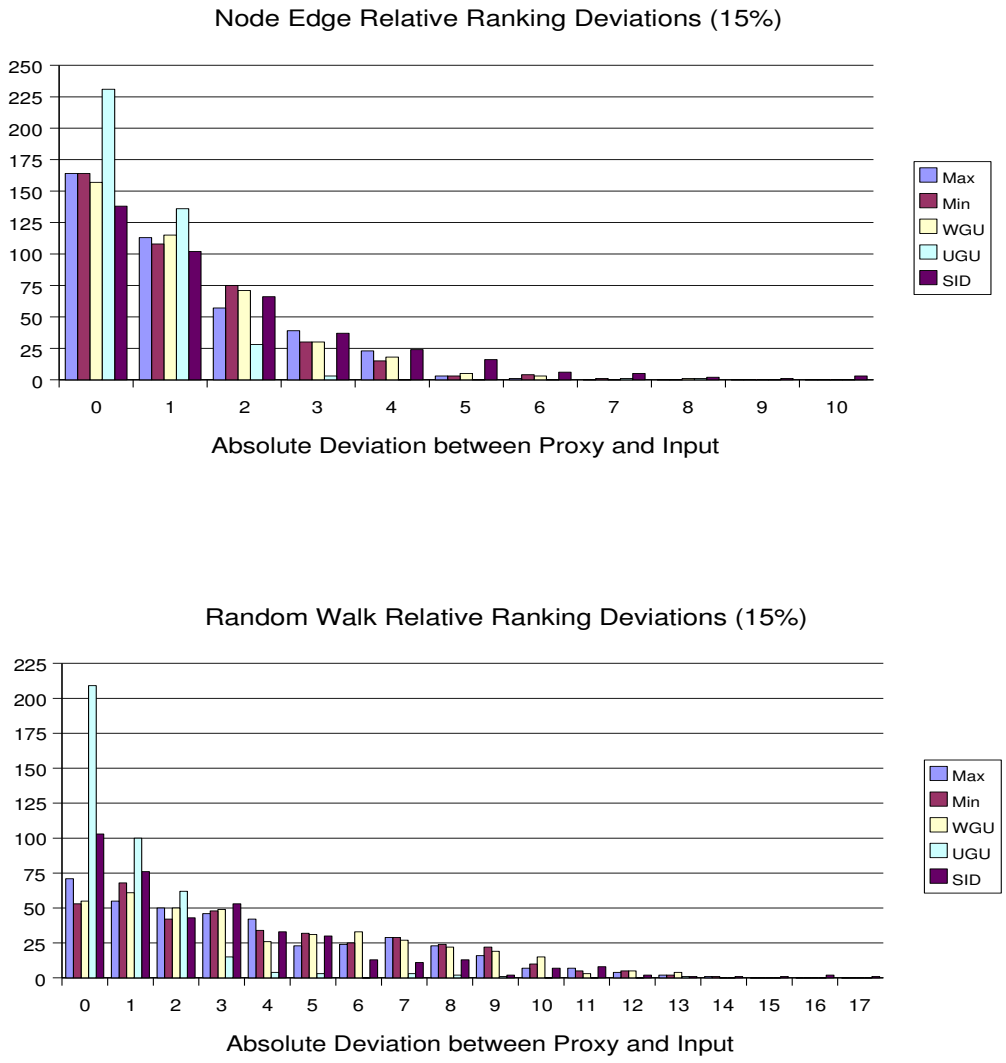


Figure 4.2: Relative changes in distance rankings from original to 10% samples. Node-edge more accurately retains relative distance rankings in its proxies.

Figure 4.3: Relative changes in distance rankings from original to 30% samples. Again we see that node-edge more accurately retains relative distance rankings in its proxies. The exception is the UGU metric which still does quite well with the random walk proxy.

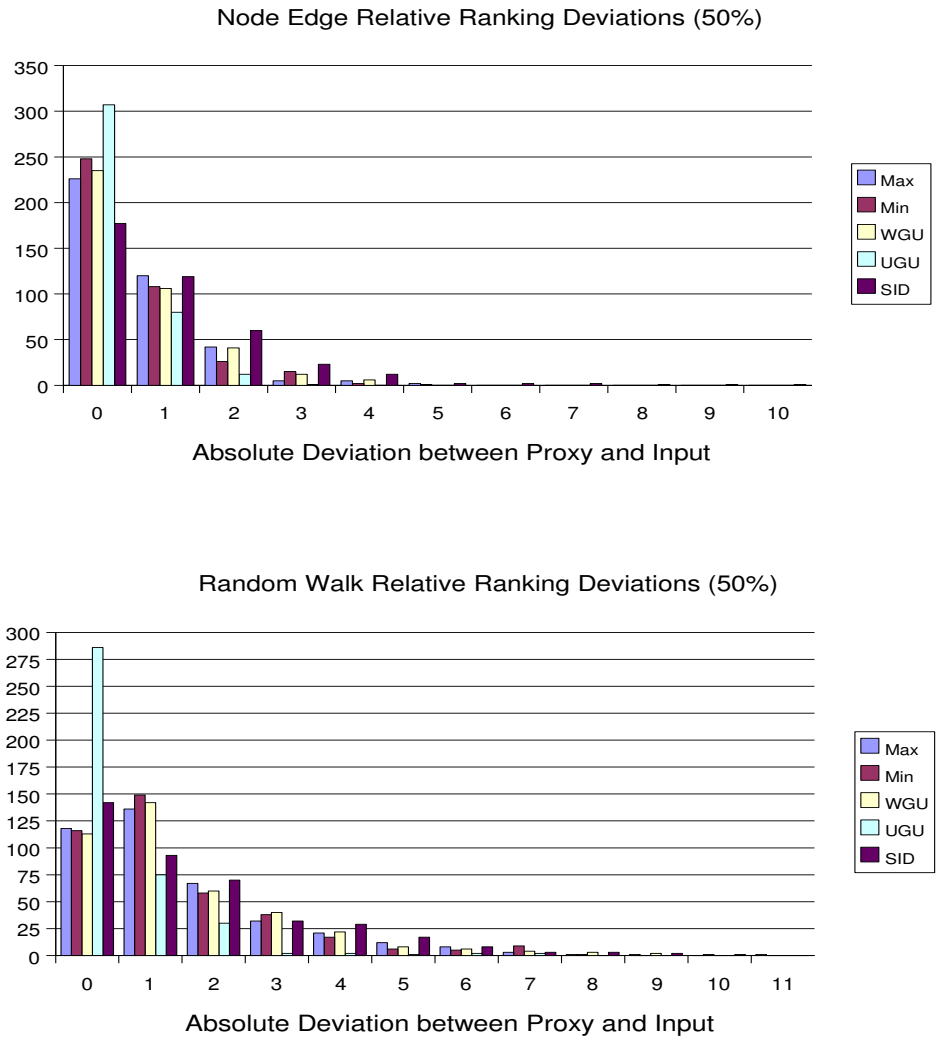


Figure 4.4: Relative changes in distance rankings from original to 50% samples. Cutting the graph's size in half produces very little change the relative rankings in distance from one graph to another.

proxies preserve the structure of the original graphs so that we may now confidently use these similarity measures in applications. As we will see in the next section, the distances between the samples result in the same classifications as the distances between the original inputs when used in a k-nearest neighbors algorithm.

4.6 Classification with Samples

Classification tasks typically attempt to classify a previously unseen input into a previously defined class. K-nearest neighbors (knn) is a common algorithm that classifies an input based on the classification of its k-closest neighbors as determined by some distance metric. In these experiments, maximum common subgraph provides the measure for similarity between two graphs, and various neighborhood sizes are used. Sample sizes of 15, 30 and 50% are used with k set to 3 and 5. Results for the 30% with k=3 are discussed specifically here.

4.6.1 Experiments

For these experiments, five classification groups are used in which to categorize each newsgroup. These author-created labels are religion, politics, computer, science, and other. These are of course arbitrary, and other current work includes using similarly created samples for clustering tasks to determine whether samples cluster into the same arbitrary neighborhoods as the original inputs. Other experiments with other classification types have been done, and the results are equally accurate. The author has assigned a default classification to each newsgroup, and the initial twenty graphs are classified according to a vote from the three closest and five closest neighbors (k=3 and k=5) as measured by the maximum common subgraph. Then samples are created for each graph, and the

same process is repeated for all pairs of sample graphs. The results from experiments with neighborhood size of three with a sample size of 30% are discussed.

4.6.2 Results

The results for classification for the 30% sample (both node-edge and random walk) of each of the original input graphs are shown in the following tables. For each, the k value for the k-nearest neighbors algorithm was set to 3 and to 5. The results for each distance metric are provided and analyzed. The potential classfications are:

- Religion
- Politics
- Tech
- Science
- Other

These categories are arbitrarily chosen, and there are naturally many other possibilities. Again, the emphasis of this work is not to come up with a necessarily more accurate classification but rather a more efficient one. For each distance metric, we will examine the sample's ability to make the same classification as the original input. Each table provides a user-assigned classification for a point of reference as well as the classifications for the original input, the node-edge sample, and the random walk sample. Predictions which do not agree with the classifications of the original input are in bold.

The results for the 30% sample are provided because they represent the results from the tradeoff of accuracy and speed. The 50% sample results are more accurate, but they do not cut the input size as drastically. The smaller

sample is faster, but is also less accurate. These are the results we would expect, but we analyze the 30% results in depth here because they provide the most insight into when and what starts to break down as the sample size grows smaller for these very large graphs. All tables referred to in the following subsection may be found at the end of this section.

Classification by Maximum Common Subgraph

Table 4.13 contains the output for the classification tasks using the maximum common subgraph distance measure. This particular metric provides the most accurate sample classifications of all of the similarity measures used in these experiments. With a k value of 3, the hybrid node-edge sample technique yields classifications that only misclassify one newsgroup with respect to the original classifications. Specifically, the Space newsgroup is classified as Religion with the original inputs but Science with the node-edge sample. Interestingly, the Science label is the user-provided label, and this leads to the question of whether samples can actually outperform the original data with respect to accuracy. However, this is left for future work as the focus here is on how well the sample mirrors the original.

With k set at 5, the maximum common subgraph still does an excellent job with only two misclassifications. One of these is again the Space newsgroup while the other is Cryptography. We will see as we go along that there are groups which can be easily categorized in multiple groups because of the diverse content, and Cryptography appears to be one such group.

The random walk sampling method does not perform as well as the node-edge sample with the maximum common subgraph. In fact, we will see that with the same sample size the random walk as implemented is consistently poorer at retaining the problem specific features most relevant to this task. While the

random walk can create a sample more quickly, at least in these experiments it appears they require a larger sample size and/or tweaking of runtime parameters such as maximum path length to perform as well as the hybrid random node-edge sample method. For the maximum common subgraph metric, random walk samples are misclassified with respect to the original classifications six out of twenty times for both k values. Five out of these six are the same for both k values, and the misclassifications are in groups that are relatively close to multiple categories of newsgroups. This again may lead to the conclusion that this implementation of random walk is not quite as robust at retaining the structure of the graph.

Classification by WGU and UGU

The WGU and UGU (Formulas 4.1 and 4.2) are variations on the maximum common subgraph metric, and their results are presented in tables 4.14 and 4.15. We would expect to see similar classification results for these distances, and in fact we see that only two node-edge samples and four random walk samples are misclassified for the WGU method. While this is not as accurate for the node-edge, the random walk actually improves with the WGU distance. For both k values, the misclassifications are for the same groups and are misclassified to the same category. We also see that some of the same groups are causing trouble – Medicine, Space, For Sale, and Mac Hardware. Again, this can be attributed not only to its close distance to several types of groups but also because of the relatively small range of distance values to other groups. This is in contrast to groups such as Religion and Atheism that are much more similar to some groups than others.

Despite accurate approximations of relative distance rankings, the UGU results in table 4.15 are very poor with respect to the classification accuracy of

the originals. The non-normalized nature of the metric appears to exacerbate its dependency on the relative sizes of graphs, and this leads not only to a majority of the originals being classified as Tech but also a majority of the samples as well. Important to note, though, is that the samples still do a very good job of classifying the newsgroups into the same groups as the originals. While such a metric would seem to need changes for improved performance with respect to classification, the samples still retain the structure in a way that the UGU classifications are nearly the same.

Classification by Minimum Common Subgraph

In table 4.13, we see the the classification results using the minimum common subgraph distances. For both k values, node-edge and random walk do a very good job of producing similarly structured samples. Only two newsgroups are misclassified with respect to the original classifications, and these are the Medicine and Space groups. Only three random walk samples are misclassified, and the usual suspects are present for this sampling method – Mac Hardware, Space, and For Sale.

Classification by Sample Intersection Distance

The sample intersection distance metric is intended as a more computationally efficient alternative to the maximum common subgraph technique because it does not need to process all of the edges from both graphs. Rather, it attempts to estimate that distance by checking a sample of that set for presence in both graphs. We see in Table 4.17 that it does indeed do a very good job of approximating the maximum common subgraph distance with respect to its use in the classification task. With k of 3, only one node-edge sample is misclassified, and it is the same as the newsgroup misclassified using the maximum common

subgraph distance. With k of 5, there is actually an improvement – only one is misclassified as compared to the maximum common subgraph's two with the node-edge samples.

We also see an improvement with the random walk samples. For k=3, only three mistakes are made. These three mistakes are interesting ones, too, because two of them (Electronics and Politics-Misc) are classified the same as the user-supplied labels while another (Christian) is misclassified as Tech. This classification is only seen in other experiments when using the UGU metric where most newsgroups were classified as such. This again begs further investigation into the ability of a sample to classify more accurately than the original data.

These results and data from other experiments using different sample sizes and classifications lead to the conclusion that sampling these large inputs can and do result in similarly structured graphs, and the maximum common subgraph between these samples produces the same general neighborhoods of proximity in the collection of graphs. By using the samples in our classification rather the original, we can significantly reduce the run time for classification as well as the space requirements by more than half. Additional experiments show that as one would expect, smaller sample sizes have more variation from the classification of the originals. Even systematically reducing the input by 75%, though, produces efficient and relatively accurate results.

		K=3			K=5		
Newsgroup	*User Label*	*Original*	*Node Edge*	*Random Walk*	*Original*	*Node Edge*	*Random Walk*
Atheism	Religion	Religion	Religion	Religion	Religion	Religion	Religion
Religion	Religion	Religion	Religion	Religion	Religion	Religion	Religion
Christian	Religion	Religion	Religion	Religion	Religion	Religion	Religion
Guns	Politics	Politics	Politics	Politics	Politics	Politics	Politics
Mideast	Politics	Politics	Politics	Politics	Politics	Politics	Politics
Politics – Misc	Politics	Religion	Religion	Religion	Religion	Religion	Religion
Graphics	Tech	Tech	Tech	Tech	Tech	Tech	Tech
IBM Hardware	Tech	Tech	Tech	Tech	Tech	Tech	Tech
Mac Hardware	Tech	Tech	Tech	**Science**	Tech	Tech	**Science**
Windows	Tech	Tech	Tech	Tech	Tech	Tech	Tech
Windows X	Tech	Tech	Tech	Tech	Tech	Tech	Tech
Cryptography	Science	Politics	Politics	Politics	Tech	**Politics**	**Politics**
Electronics	Science	Tech	Tech	**Other**	Tech	Tech	Other
Medicine	Science	Science	Science	**Religion**	Science	Science	**Religion**
Space	Science	Religion	**Science**	**Science**	Tech	***Science***	**Science**
Autos	Other	Other	Other	Other	Other	Other	Other
Motorcycles	Other	Other	Other	Other	Other	Other	Other
For Sale	Other	Tech	Tech	**Science**	Tech	Tech	**Science**
Baseball	Other	Other	Other	**Science**	Other	Other	**Science**
Hockey	Other	Other	Other	Other	Other	Other	Other

Table 4.13: Classification of Newsgroups by Max Common Subgraph with $K = 3$ and $K = 5$

		K=3			K=5		
Newsgroup	*User Label*	*Original*	*Node Edge*	*Random Walk*	*Original*	*Node Edge*	*Random Walk*
Atheism	Religion	Religion	Religion	Religion	Religion	Religion	Religion
Religion	Religion	Religion	Religion	Religion	Religion	Religion	Religion
Christian	Religion	Religion	Religion	Religion	Religion	Religion	Religion
Guns	Politics	Politics	Politics	Politics	Politics	Politics	Politics
Mideast	Politics	Politics	Politics	Politics	Politics	Politics	Politics
Politics – Misc	Politics	Religion	Religion	Religion	Religion	Religion	Religion
Graphics	Tech	Tech	Tech	Tech	Tech	Tech	Tech
IBM Hardware	Tech	Tech	Tech	Tech	Tech	Tech	Tech
Mac Hardware	Tech	Tech	Tech	**Science**	Tech	Tech	**Science**
Windows	Tech	Tech	Tech	Tech	Tech	Tech	Tech
Windows X	Tech	Tech	Tech	Tech	Tech	Tech	Tech
Cryptography	Science	Politics	Politics	Politics	Politics	Politics	Politics
Electronics	Science	Tech	Tech	**Other**	Tech	Tech	**Other**
Medicine	Science	Science	**Politics**	Science	Science	**Politics**	Science
Space	Science	Religion	**Politics**	**Science**	Religion	**Politics**	**Science**
Autos	Other	Other	Other	Other	Other	Other	Other
Motorcycles	Other	Other	Other	Other	Other	Other	Other
For Sale	Other	Tech	Tech	**Science**	Tech	Tech	**Science**
Baseball	Other	Other	Other	Other	Other	Other	Other
Hockey	Other	Other	Other	Other	Other	Other	Other

Table 4.14: Classification of Newsgroups by WGU with $K = 3$ and $K = 5$

		K=3			K=5		
Newsgroup	*User Label*	*Original*	*Node Edge*	*Random Walk*	*Original*	*Node Edge*	*Random Walk*
Atheism	Religion	Religion	Religion	**Tech**	Religion	Religion	**Tech**
Religion	Religion	Religion	Religion	**Tech**	Religion	Religion	**Tech**
Christian	Religion	Tech	Tech	Tech	Tech	Tech	Tech
Guns	Politics	Tech	Tech	Tech	Tech	Tech	Tech
Mideast	Politics	Tech	Tech	Tech	Tech	Tech	Tech
Politics – Misc	Politics	Religion	**Tech**	**Tech**	Religion	**Tech**	**Tech**
Graphics	Tech	Tech	Tech	Tech	Tech	Tech	Tech
IBM Hardware	Tech	Tech	Tech	Tech	Tech	Tech	Tech
Mac Hardware	Tech	Tech	Tech	**Other**	Tech	Tech	**Other**
Windows	Tech	Tech	Tech	Tech	Tech	Tech	Tech
Windows X	Tech	Tech	Tech	Tech	Tech	Tech	Tech
Cryptography	Science	Tech	Tech	Tech	Tech	Tech	Tech
Electronics	Science	Tech	Tech	Tech	Tech	Tech	Tech
Medicine	Science	Tech	Tech	Tech	Tech	Tech	Tech
Space	Science	Tech	Tech	Tech	Tech	Tech	Tech
Autos	Other	Tech	**Other**	Tech	Tech	**Other**	Tech
Motorcycles	Other	Tech	Tech	Tech	Tech	Tech	Tech
For Sale	Other	Tech	Tech	Tech	Tech	Tech	Tech
Baseball	Other	Tech	Tech	Tech	Tech	Tech	Tech
Hockey	Other	Tech	**Other**	Tech	Tech	**Other**	Tech

Table 4.15: Classification of Newsgroups by UGU with $K = 3$ and $K = 5$

		K=3			K=5		
Newsgroup	*User Label*	*Original*	*Node Edge*	*Random Walk*	*Original*	*Node Edge*	*Random Walk*
Atheism	Religion	Religion	Religion	Religion	Religion	Religion	Religion
Religion	Religion	Religion	Religion	Religion	Religion	Religion	Religion
Christian	Religion	Religion	Religion	Religion	Religion	Religion	Religion
Guns	Politics	Politics	Politics	Politics	Politics	Politics	Politics
Mideast	Politics	Politics	Politics	Politics	Politics	Politics	Politics
Politics – Misc	Politics	Religion	Religion	Religion	Religion	Religion	Religion
Graphics	Tech	Tech	Tech	Tech	Tech	Tech	Tech
IBM Hardware	Tech	Tech	Tech	Tech	Tech	Tech	Tech
Mac Hardware	Tech	Tech	Tech	**Science**	Tech	Tech	**Science**
Windows	Tech	Tech	Tech	Tech	Tech	Tech	Tech
Windows X	Tech	Tech	Tech	Tech	Tech	Tech	Tech
Cryptography	Science	Politics	Politics	Politics	Politics	Politics	Politics
Electronics	Science	Tech	Tech	Tech	Tech	Tech	Tech
Medicine	Science	Science	**Politics**	Science	Science	**Politics**	Science
Space	Science	Religion	**Politics**	**Science**	Religion	**Politics**	**Science**
Autos	Other	Other	Other	Other	Other	Other	Other
Motorcycles	Other	Other	Other	Other	Other	Other	Other
For Sale	Other	Tech	Tech	**Science**	Tech	Tech	**Science**
Baseball	Other	Other	Other	Other	Other	Other	Other
Hockey	Other	Other	Other	Other	Other	Other	Other

Table 4.16: Classification of Newsgroups by Min Common Supergraph with $K = 3$ and $K = 5$

		K=3			K=5		
Newsgroup	*User Label*	*Original*	*Node Edge*	*Random Walk*	*Original*	*Node Edge*	*Random Walk*
Atheism	Religion	Religion	Religion	Religion	Religion	Religion	Religion
Religion	Religion	Religion	Religion	Religion	Religion	Religion	Religion
Christian	Religion	Religion	Religion	**Tech**	Religion	Religion	**Tech**
Guns	Politics	Politics	Politics	Politics	Politics	Politics	Politics
Mideast	Politics	Politics	Politics	Politics	Politics	Politics	Politics
Politics – Misc	Politics	Religion	Religion	**Politics**	Religion	Religion	**Politics**
Graphics	Tech	Tech	Tech	Tech	Tech	Tech	Tech
IBM Hardware	Tech	Tech	Tech	Tech	Tech	Tech	Tech
Mac Hardware	Tech	Tech	Tech	Tech	Tech	Tech	Tech
Windows	Tech	Tech	Tech	Tech	Tech	Tech	Tech
Windows X	Tech	Tech	Tech	Tech	Tech	Tech	Tech
Cryptography	Science	Tech	Tech	Tech	Tech	Tech	Tech
Electronics	Science	Tech	Tech	**Science**	Tech	Tech	**Science**
Medicine	Science	Science	Science	Science	Science	Science	Science
Space	Science	Tech	**Science**	Tech	Tech	**Science**	Tech
Autos	Other	Other	Other	Tech	Other	Other	**Tech**
Motorcycles	Other	Other	Other	Tech	Other	Other	**Tech**
For Sale	Other	Tech	Tech	Tech	Tech	Tech	Tech
Baseball	Other	Other	Other	Tech	Other	Other	**Tech**
Hockey	Other	Other	Other	Other	Other	Other	Other

Table 4.17: Classification of Newsgroups by SID with k=3,k=5

4.7 Conclusions

This chapter has shown that the use of random samples of large inputs for data mining processes can successfully reduce the time and memory requirements while providing the same results for tasks such as classification. Very large graphs were processed to determine their similarity using five different distance metrics, including one introduced in this work. Two sampling methods (a hybrid node-edge introduced here as well as a random walk) were used to create more compact representations of the data, and the same distance metrics were used to calculate the similarity of the samples. It was shown that both sampling techniques do a very good job of retaining the aggregate structure of the graphs as the relative rankings of the distances between graphs varied very little from original data to sample and up to a 95% accuracy rate with classification prediction.

K-nearest neighbors classification also proved to be much faster with very little loss in accuracy. With the exception of the UGU classifications, all of the distance metrics used did an excellent job of making the same classifications as those made on the original input. For the maximum common subgraph and sample intersection distance metrics, only one misclassification of the node-edge sample was made (with respect to the classification of the original inputs). These improvements in efficiency are due to the ability of sampling to retain the properties of large inputs such as graphs created from web content. As inputs to mining tasks grow larger and larger, such data reduction techniques become more and more important to maintain efficient computation.

Chapter 5

Clustering Large Graphs

5.1 Introduction

Clustering is another common data mining task used widely to create groups automatically from inputs based on their similarity. That is, given some set of inputs and a number of clusters, the process assigns the inputs into their most-likely grouping according to some distance metric. With large graphs, traditional mining algorithms such as k-means can be modified to use graph distance metrics for clustering [2]. Inputs to such problems continue to grow in size, and techniques to reduce the processing and storage complexity of such inputs is just as important.

As we have already seen in the previous chapter, web content data are easily and robustly modeled using graphs. By storing not only elements of the input (words) but also relationships between them as encoded by their proximity, a richer representation is possible. Such expressiveness can enable more accurate data mining, but it can come with a price. Sampling techniques aim to reduce the size of inputs with minimal loss of representational power.

In this chapter, data collected from a web-accessible news site are clustered using a k-means algorithm where graphs cluster around a median. This median is defined as the graph in the cluster with the shortest total distance to all other graphs in that cluster, and it is meant to be the most representative graph in a given cluster. Given news articles from over forty subcategories, we will see that

we can accurately cluster them into seven broader categories using large input proxies.

Previous experiments have shown the hybrid node-edge sampling method to more accurately retain the relative similarities of graphs as compared to the random walk as implemented, and so it is the sampling method of choice for these clustering experiments. Also, the three most successful distance metrics are used – maximum common subgraph, minimum common supergraph, and the sample intersection distance.

5.2 Related Work

There are several approaches to clustering, but they all rely on some notion of similarity or distance between inputs. There exist both top-down and bottom-up methods that assume all inputs are in the same cluster and break them up into subsets or assume that all inputs are in their own clusters and combine them into supersets [59]. Agglomerative or divisive methods fall into the category of hierarchical clustering methods [60], but the approach used for clustering in this chapter is an iterative partitioning method known as k-means. An excellent overview of various clustering techniques is provided in [61].

There exist many types of inputs for clustering which themselves contain an array of element types from continuous and discrete numeric value to nominal and other ordinal values. Graph clustering is often thought of as partitioning a single graph into distinct clusters, but this chapter focuses on clustering multiple graphs into distinct clusters. Other work has focused on this task of clustering several graphs [51, 29, 2], and it has been shown to be an expensive one because of the cost in calculating the distance or similarity between different graphs. SLIP reduces both the space and time requirements for the clustering task.

5.2.1 Clustering Web Content

Clustering is a data mining technique which attempts to create categories or groups into which inputs are separated. Given some number of inputs, the problem involves coming up with a way to decide which should be grouped together according to some sort of similarity metric. Graphs are very expressive, and have been shown to outperform other means of web content mining representation such as various vector models [2]. Indeed, the larger the graph is allowed to grow to represent the document, the better it performs. This increase in size comes with a cost, and the sampling techniques used here attempt to take advantage of the expressiveness of the graph while greatly reducing the computational expense.

A common algorithm used to solve the clustering problem is k-means, and this chapter focuses on extending an adapted k-means clustering algorithm on graph data introduced in [2]. Given some number of groups k into which we want to separate the inputs, the algorithm repeats a process of assigning inputs to the cluster in which its representative is closest. For graph data, these experiments use the notion of a median graph of group to select a representative of the cluster. Similar to the k-mediod clustering in the vector model, the median graph is the graph that minimizes distance to all other graphs in that cluster and is therefore considered representative. In addition, a more globally optimal seeding of initial medians is used in improve the clustering performance. More details are provided in section 5.6. The data set is overviewed in the next section, and a review of the sampling methods and distance metrics is provided in Section 5.4 and Section 5.5.

5.3 Data Sets

The data for these experiments come the Yahoo! News website [62]. Like many other news sites, Yahoo! uses a hierarchical structure to organize the news articles. For each broader category, several more specific subcategories exist. Each subcategory contains as few as five and as many as one hundred recent news articles on the subject. News articles across 7 general and 42 subcategories were downloaded from the Yahoo! News website. Each category along with an abbreviation is listed in Table 5.1, and the subcategories along with its Yahoo! general category is listed in Table 5.2.

Category	*Abbreviation*
United States	US
Business	B
World	W
Entertainment	E
Sports	Sp
Technology	T
Science	Sc

Table 5.1: News Article Categories from Yahoo! News

Each subcategory of data used in these experiments contains as few as 8 and as many as 40 articles. For each subcategory, the articles are aggregated into a single input, and these inputs are the basis for the graphs that are generated for clustering. The same process used previously to generate the graphs is used. Stop words are removed from the text, and nodes are created from the remaining words. Edges are added according to proximity. The proximity parameter for these experiments is 3, but this can be adjusted to create more or less dense graphs. Because the graphs are relatively small compared to the newsgroup data, sample sizes of 50% are used here.

Education (US)	Movies (E)	Football (Sp)
Religion (US)	Music (E)	Basketball (Sp)
Politics (US)	Television (E)	Baseball (Sp)
Crimes/Trials (US)	Industry (E)	Soccer (Sp)
	Books (E)	Hockey (Sp)
Markets (B)	Celebrity (E)	College (Sp)
Earnings (B)	Fashion (E)	Tennis (Sp)
Economy (B)		Golf (Sp)
		Motor Sports (Sp)
Mideast (W)	Personal Tech (T)	
Europe(W)	Communications (T)	
Latin America(W)	Software (T)	Weather (Sc)
Africa (W)	Apple/Mac (T)	Space (Sc)
Asia (W)	Open Source (T)	Animals (Sc)
Asia (W)	Internet (T)	Biotech (Sc)
Canada (W)		Energy (Sc)
Australia (W)		Environment (Sc)

Table 5.2: The 41 news subcategories grouped into the seven general categories from Yahoo! News. The inputs will be clustered into seven groups in order to see how well they recreated the Yahoo! assigned clusterings.

All inputs (graphs composed of all articles of a subcategory) will be clustered into seven groups, and the performance of the k-means algorithm on both the original inputs and the samples will be compared to the Yahoo! categories. One might imagine we would hope to see perfect clustering with the original graphs as well as with the proxies. That is, we would like to see that all of the Sports subcategories are grouped to together while all of the Politics subcategories are grouped together. However, what we are again aiming for with these experiments is to reproduce the clusterings of the original inputs with the proxy inputs. The seven categories provided are user-supplied, and they may or may not represent the best actual labels for the individual articles. Also, it is frequently the case that articles belong to multiple categories because of their subject. Such issues are important, but they are reserved for future work.

5.4 Sampling Methods

The method for sampling the input graphs is the same hybrid node-edge random sampling technique defined previously as a two-step process that randomly selects some number of randomly selected edges from the input graph's edge list (see Algorithm 4). For each edge selected, the source and target nodes are added to the sample graph. This is repeated until the required number of nodes is reached. Then, for each node in this subgraph, each adjacency list is examined. If the destination node for an edge in this list is not in the sample's node list, that edge is removed from the source node's adjacency list. This method performs well for maintaining the structure for different types of graphs, and we will see that it does well for the smaller graphs created from the news data.

5.5 Distance Metrics

Several classes of distance measures for graphs exist, but the ones used in these clustering experiments are the maximum common subgraph, minimum common supergraph, and the sample intersection distance. The sample intersection distance is meant to be a rough estimate of the maximum common subgraph metric while the minimum common supergraph measure incorporates notions of the intersection and union of the graphs. For the much larger graphs created from the newsgroup data, these three provided the most variety in performance while maintaining accuracy and so have been chosen as the metrics for the clustering. More details on each of these may be found in section 4.4 in chapter 4.

Again we are initially concerned with the ability of the distance metrics to maintain relative distances between the various subcategories of news articles. Changes in relative distance between the original inputs and the proxies are presented in table 5.5, and a visualization of this data is provided in figure 5.1. Full distance data for each metric is provided in Appendix B.

We see that the majority of the changes in distance rankings are relatively small for all distance metrics. While minimum common supergraph has the most pairs of inputs whose relative distance rankings does not change, maximum common subgraph seems to preserve relative ordering better overall. The sample intersection distance metric has a much thicker tail as it does not do as accurate of a job in preserving relative ordering in the proxies. While all metrics do a good job of maintaining distances, we would expect to see the maximum common subgraph and minimum common supergraph allow more accurate clustering of the proxies based on these rankings. With evidence that the graph proxies maintain the relative distances between each other, we may now compare the clusterings of the proxies as compared to the clusterings of the original inputs.

Ranking Change	Max	Min	SID	Ranking Change	Max	Min	SID
0	378	387	262	15	5	6	20
1	350	328	245	16	4	10	16
2	271	238	176	17	4	4	19
3	190	179	179	18	1	4	12
4	125	158	153	19	1	2	14
5	109	111	102	20	1	2	8
6	80	80	94	21	1	1	10
7	65	71	92	22	1	0	4
8	51	41	79	23	1	0	3
9	30	34	72	24	0	1	8
10	21	31	56	25	0	1	4
11	28	29	41	26	0	0	2
12	26	19	37	27	0	0	2
13	6	15	30	28	0	0	1
14	15	12	21				

Table 5.3: Changes in Relative Distance Ranking from Original to Proxy

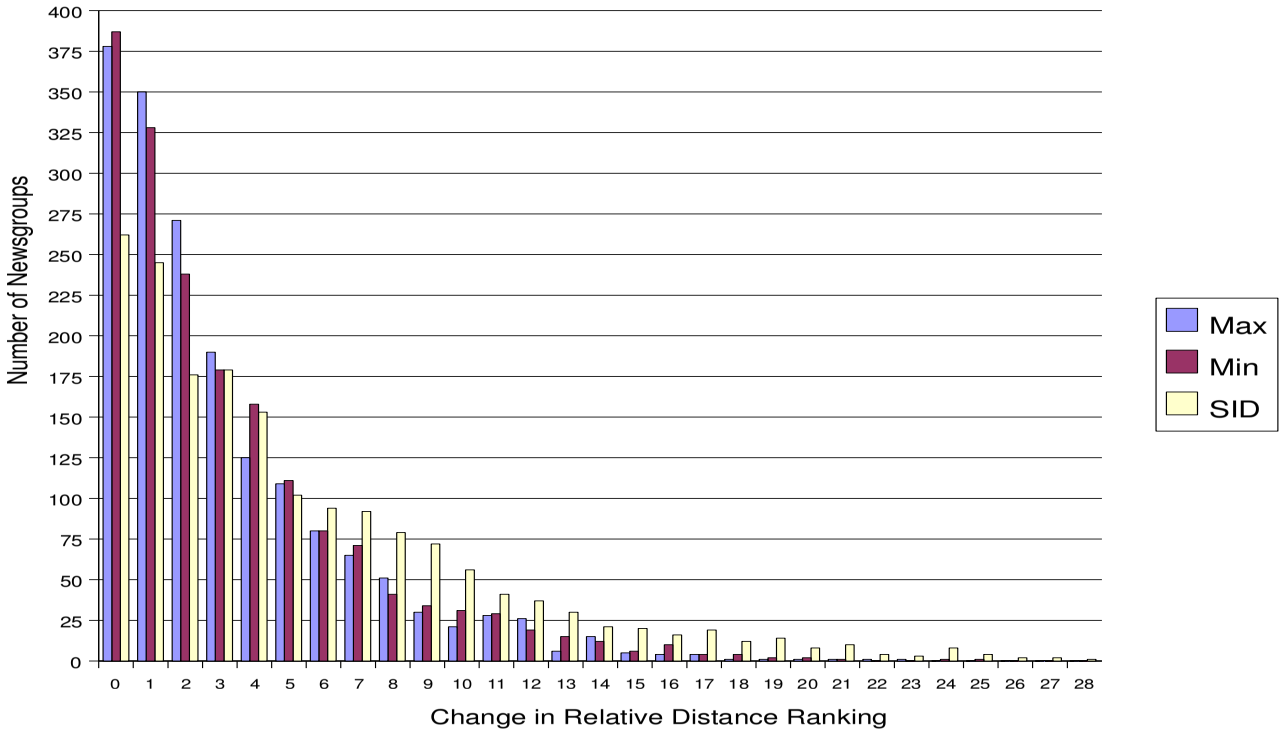


Figure 5.1: Changes in relative distance from original to 50% proxy. Overall, the maximum common subgraph metric appears to maintain ordering the best for the relatively small graphs created from the news articles. We see that SID begins to break down in its performance as the inputs become smaller.

5.6 Clustering Experiments

Clustering is performed using a version of *k*-means that relies on the notion of a graph median. This median of a cluster is defined as the graph whose total distance to all other graphs in the cluster is the minimal of all such graphs in the cluster.

$$\forall g \epsilon G \sum_{i}^{n} d(g, g_i) \tag{5.1}$$

Other notions of the center of a cluster of graphs exist such as finding the mean graph of a cluster. This could be defined as the largest subgraph of all graphs in the cluster, but finding such a graph becomes inordinately expensive as the number of inputs and cluster sizes grow. The graph median is used because of its relatively efficient calculation compared to such a process.

The initial clusters in *k*-means are often randomly populated, and the clusters are allowed to evolve from there. Such a random assignment often leads to local optima because the best representative graphs are not chosen as initial medians. The algorithm used here is a graph-based version of the global *k*-means method introduced by [63]. This attempts to find the best seeds as medians for the clusters before the clustering begins in earnest, and it has been shown to provide as good and often better clustering results.

Using this definition of median and various distance metrics $d(g, g_i)$ between two graphs, the *k*-means algorithm used is in Algorithm 6:

As mentioned previously, the *k*-means algorithm can suffer when the clusters are initialized randomly. Performance can vary from run to run, and global optima are not guaranteed. A global *k*-means algorithm adapted to graph clustering attempts to find the best median graphs for each cluster before the clustering process begins to iterate. It is presented in algorithm 7.

Algorithm 6: K-means for Graph Clustering

Input: Set of Graphs G, k number of clusters (C_k)
Output: Set of clusters C_k containing all elements of G
Find initial cluster medians (Algorithm 7)
for *cluster* $c \epsilon C_k$ **do**
  select random graph g_i from G as median of cluster
end
while *some* g_i *assigned to new cluster* **do**
  for $g_i \epsilon G$ **do**
    Assign to cluster c to which distance to median is shortest
  end
  for *each* $c \epsilon C_k$ **do**
    Calculate new median of cluster
  end
end
return C_k

Algorithm 7: Initializing clusters with optimal seeds

Input: Set of Graphs G, k number of clusters (C_k)
Output: Set of cluster medians C_k containing median seed graphs for all clusters
Initialize median of first cluster to median of all graphs in G
for *successive cluster* $c \epsilon C_k$ **do**
  for *each graph* $g \epsilon G$ **do**
    if $\sum_{i}^{n} max(d_{k-1}^{j} - d(i,j)^2, 0) >$ *all other* g_i **then**
      mark g_i
    end
  end
  set marked g_i as median for cluster k
end

To measure the performance of clustering, two measures are used that attempt to measure how well the clusters match the original groupings as well as how compact and well-separated the clusters are. First the Rand index measures the quality of the clusters by comparing the number of agreements and disagreements [64]. For all pairs of inputs, if the two graphs are in a cluster when they are supposed to be or not in a cluster when they are not supposed to be, then it counts as an agreement. Otherwise, if two inputs end up in a cluster together when they were not originally or end up in different clusters when they were together originally, it counts as a disagreement. The index is given in formula 5.2.

$$R_i = \frac{A}{A + D} \tag{5.2}$$

The Dunn index provides a sense of how well the clusters are constructed by considering how compact and separate they are [65]. In formula 5.3, d_{min} is the minimum distance between the two closest inputs not in the same cluster while d_{max} is the maximum distance between two inputs in the same cluster.

$$D_i = \frac{d_{min}}{d_{max}} \tag{5.3}$$

Both indexes are provided for both the original inputs as well as the 50% samples to compare performance.

5.7 Results

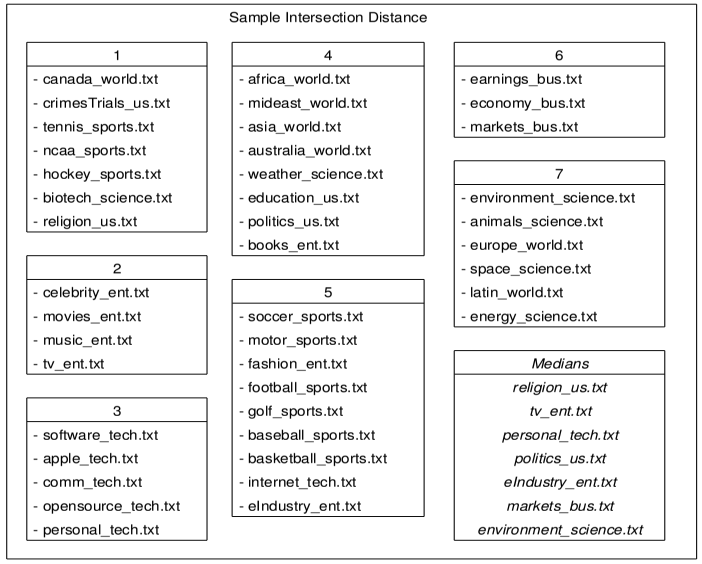
The clusterings produced are presented in Tables 5.5 and 5.6. These are based on the original graphs as well as 50% node-edge samples using three distance metrics. The *k* value for all experiments is set to 7 as this is the number of Yahoo! general news categories. Further work could investigate the optimality

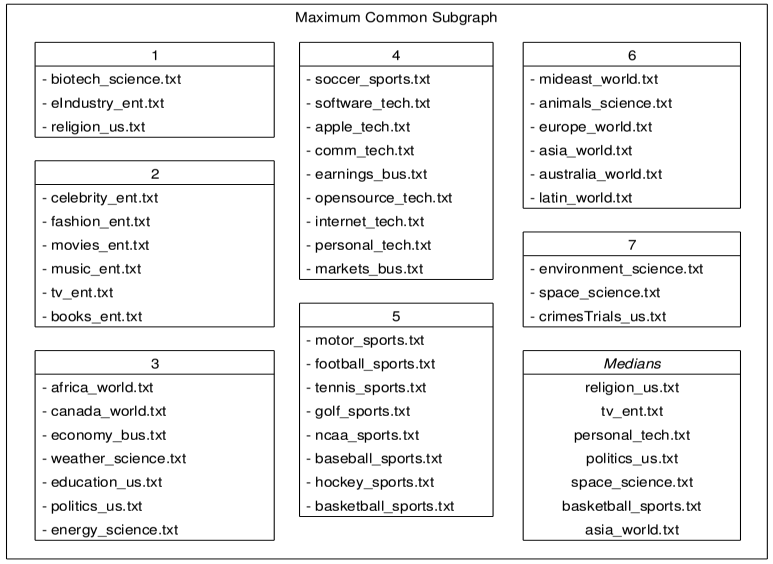
of different cluster sizes with respect to various measures of clustering performance.

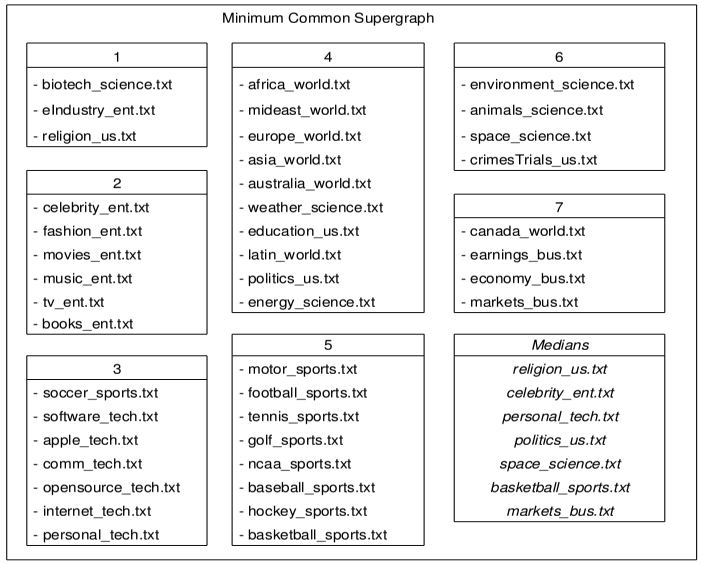
Both the original and the sample methods do an excellent job of clustering. The indexes summarized in Table 5.4 indicate that not only do the original graphs accurately represent the information contained in the news articles but that the proxies themselves also do a very good job in retaining that structure with very little if any sacrifice in clustering ability. The minimum common supergraph performs the best with the original input, but the maximum common subgraph's performance with the sample barely beats out the minimum common supergraph for best performance with respect to the Rand index. We also see that while the sample intersection distance does not perform poorly, it does do measurably worse than the other metrics. This can be attributed at least in part to the size of the input graphs.

Distance Metric	Rand		Dunn	
	Original	Sample	Original	Sample
Max Common Subgraph	0.9047	0.9005	0.8619	0.8923
Min Common Supergraph	0.9125	0.8955	0.9119	0.9359
Sample Intersection Distance	0.8487	0.8676	0.8812	0.9259

Table 5.4: Cluster performance indexes for Yahoo! News graphs and their proxies

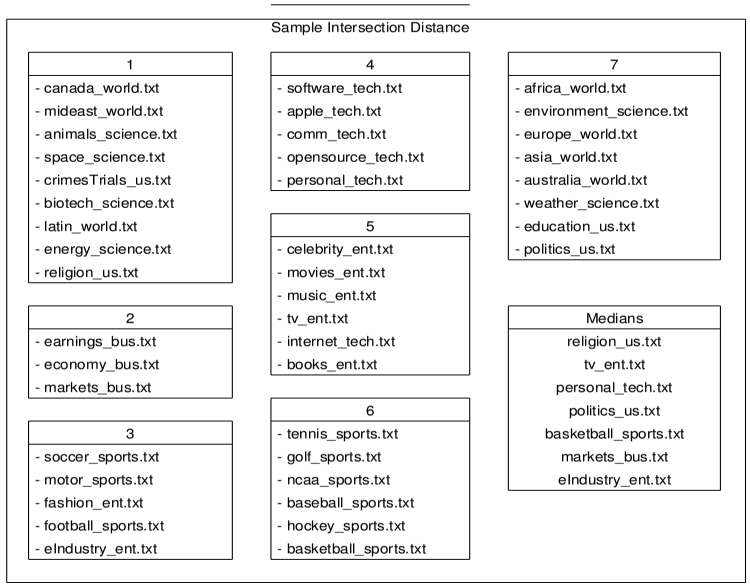


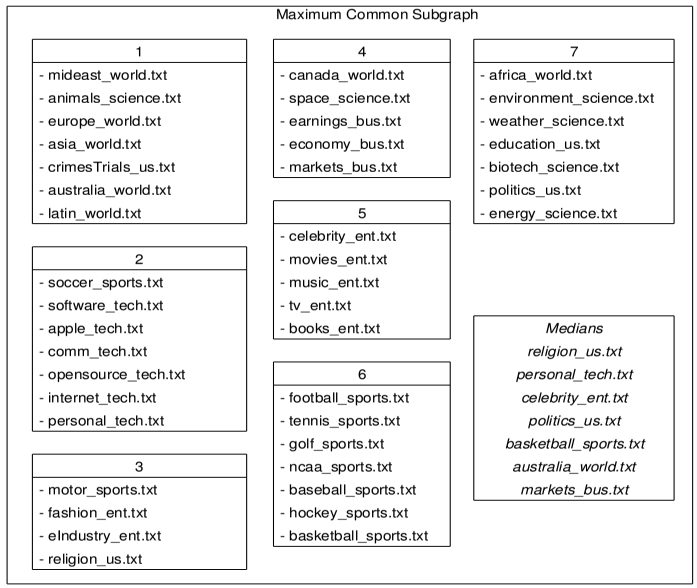


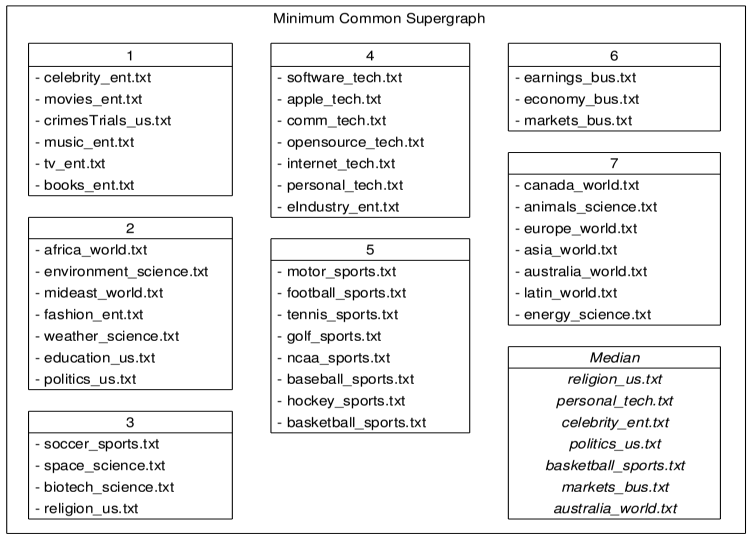


	Rand	Dunn
Max Common Subgraph	**0.9047**	**0.8619**
Min Common Supergraph	**0.9125**	**0.9119**
Sample Intersection Distance	**0.8487**	**0.8812**

Table 5.5: Clusterings produced with global k-means on original inputs. These are the clusterings produced from the original inputs and consequently the ones we hope to emulate with the proxy clusterings. The medians for each cluster are listed, and the Rand and Dunn indexes for each distance metric are also provided.







	Rand	Dunn
Max Common Subgraph	**0.9005**	**0.8923**
Min Common Supergraph	**0.8955**	**0.9359**
Sample Intersection Distance	**0.8676**	**0.9259**

Table 5.6: Clusterings produced with global k-means on 50% samples. For each distance metric, the contents of all seven clusters are provided as well as the Rand and Dunn indexes. The proxies are clustered very similarly to the originals, and the indexes indicate they do an excellent job while using significantly less space.

We saw very good success with the SID metric in the classification task of the newsgroups, but this may be attributed to the much larger size of those graphs. Because SID is meant as a fast approximation of the maximum common subgraph, we might expect it to perform better on more robust graphs. With these inputs, however, the graphs are significantly smaller, and the randomly chosen edges from these sets of much smaller edge lists do not seem to perform as well. We might expect a higher sample rate in the sample intersection distance metric to provide better results.

In all three cases, the proxies provide tighter clustering according to the Dunn index. Because this index is relative to the distances between the graphs, it must be interpreted with care. In tables 5.7 and 5.8, we see the pairs of graphs whose distances were used to create the Dunn index for both the originals and the proxies. For the originals, we see that the closest graphs not in the same cluster are actually graphs that are in the same Yahoo! categories. The Movies and Entertainment Industry categories are placed into separate categories with the original inputs, but two of the proxies actually group them together.

Distance Metric	Closest Graphs Not in Same Cluster	Most Distant in Same Cluster
Max Common	(Movies, eIndustry)	(Soccer, Open Source)
Min Common	(Movies, eIndustry)	(Soccer, Open Source)
Sample Intersection	(Movies, eIndustry)	(Tennis, Biotech)

Table 5.7: Closest and Most Distant Original Graphs Used in Dunn Index

It is interesting that the samples appear to provide tighter clustering. It is difficult to make an absolute comparison between the original inputs and the proxies with respect to the Dunn index values, though, because the distance values between these two sets of differently sized graphs are different. That is,

generally speaking, the distances between any two proxies is larger than the distance between the same two original inputs. This slight change makes it more difficult to compare, but we can still see that the proxies provide good clustering. In fact, they may even provide better clustering in some cases as evidenced by the closest graphs not in the same cluster. Extensions of this work might analyze the ability of such proxies not only to improve efficiency but accuracy (discussed in section 6.2).

Distance Metric	Closest Graphs Not in Same Cluster	Most Distant in Same Cluster
Max Common	(Comm, Earnings)	(Soccer, Open Source)
Min Common	(Comm, Earnings)	(Canada, Latin)
Sample Intersection	(Movies, eIndustry)	(Mideast, Biotech)

Table 5.8: Closest and Most Distant Proxy Graphs Used in Dunn Index

5.8 Conclusions

With graphs ranging in size from hundreds to tens of thousands of nodes (and several times that many edges) modeling web content, random sampling helps to reduce the computational requirements with no great loss in accuracy for various data mining tasks. Computing the similarity between two graphs requires an examination of the node and edge sets, and even for a labelled graph it remains an expensive calculation. Using SLIP, the relative distances between graphs is preserved in much smaller proxies for use in clustering. Using a collection of Yahoo! News articles, it has been shown that the proxies cluster virtually as well as the original inputs. For the sample intersection distance metric, performance actually improved. Such effective reduction in processing requirements is crucial in handling the large amounts of data we face on the web and from other sources.

Chapter 6

Conclusion

6.1 Summary

Proxies for large inputs are extremely useful in situations where processing the original inputs is simply infeasible because of time or storage requirements. Sampling for large input proxies (SLIP) provides a general framework to reduce the size of inputs while maintaining inherent structure as reflected in problem specific features. A comparison of the enumerated PSF's for both the original and the proxy inputs of two specific input types across four different applications shows that SLIP is able to create proxies that retain identifying structure very well.

First, sortable permutations of various sizes and in various stages of sortedness were created and sampled. Various features of sortedness such as the number of inversions were compared to show that the proxies share a similar structure with respect to the sorting problem. The proxies were then used in two related sorting problems – sorting algorithm selection and Shellsort increment set selection. For both, SLIP produces proxies that do a good job of predicting the best algorithm or increment set to use for a given input. Both problems also shared an interesting range of inputs where SLIP occasionally predicted incorrectly. This class of inputs was typically in between sorted and random, and the inaccuracy stems from the varying overhead of sorting algorithms for different sized inputs. For such algorithms, speed is as much a function of size as the problem specific features, and SLIP produces small enough proxies that poor

predictions sometimes occur. Overall, though, the best algorithms and increment sets are predicted accurately because the features of sortedness are maintained in the proxies.

SLIP also does an excellent job of preserving problem specific features related to classification and clustering of large graphs. Specifically, various measures of distance were calculated for graphs created from two different corpora of web content. In both cases, nodes were created based on words, and these nodes were linked by edges based on word proximity. Two distinct data sets were used to test SLIP's robustness with respect to graph size for the classification and clustering tasks. The first data set consisted of the text of twenty newsgroups, and they produced very large graphs on the order of tens of thousands of nodes. The second set consisted of the text of collected news articles from 41 categories of Yahoo! News. These 41 are hierarchically categorized into seven more general categories.

For the very large graphs of newsgroup data, two sampling methods were used to create proxy inputs of various sizes. For each, five different metrics of graph distance were calculated between each original graph and between each proxy graph. Larger sample sizes naturally produced more accurate retention of these relative distances, but even relatively small proxy graphs did a very good job of approximating the problem specific feature. For the 30% sample, classification was extrememly accurate as compared to the same classification for the original inputs using the k-nearest neighbors algorithm. Maximum common subgraph and its related metrics performed extremely well while all of the proxies were classified accurately with a significant gain in performance.

The Yahoo! News graphs are smaller with respect to graph size, but they cover many more categories and were more dynamic as their major subjects might change from week to week. Proxies were created, and the relative graph dis-

tances varied little from original inputs to proxies. Both the original and the SLIP proxies were clustered using the k-means algorithm, and the samples were very accurate considering the size reduction. Several distance metrics were tested with respect to different clustering performance indexes, and the samples sometimes even outperformed the original outputs. For both the very large graphs created from the newsgroup data and the smaller graphs created from the larger variety of Yahoo! News articles, SLIP created reliable proxies that could be used in classification and clustering tasks very effectively.

6.2 Future Work

There are several potentially interesting extensions of this work. For algorithm selection, there are several problems with a variety of solutions which vary in performance depending on problem specific features. For example, potential applications in selecting the best SAT [1] solver heuristics were examined, but the sampling techniques used proved not robust enough to capture the problem specific features of the SAT problem which determine algorithm performance. SLIP appears to be a good candidate for problems whose inputs are not brittle in the sense that somewhere in a very large input is a single element or small set of elements which ultimately determine how hard the instance is. These sorts of problem specific features are more difficult to capture with random sampling because sampling is not necessarily the best approach to find PSF's not inherent throughout an input's structure but rather isolated to one element. Other algorithm selection problems include choosing machine learning techniques such as different classifiers that take the same types of input. SLIP could be used to

[1] The satisfiability problem of deciding whether a particular Boolean expression has a satisfying truth assignment

determine at run-time which technique could provide the most accurate results before exhaustive training and testing on full inputs.

Sampling issues include a thorough theoretical analysis of sample size where size can take on two meanings. First, how large should the proxy be as a percentage of the original input? Second, how many such proxies must be taken to achieve a certain confidence in a variety of problem specific features? Another sampling issue is the application of stratification in various sampling techniques that may potentially reduce the variance and necessary sample sizes of different parts of the population (here roughly defined as a single large input).

For web content mining, SLIP already does an excellent job of approximating the classifications and clusterings of the original inputs. It was noted, however, that there appeared to be cases where the samples may have actually outperformed the original data. In these instances, SLIP may actually help to reduce noise in inputs to filter them into more representative inputs whose similarities and differences are even more pronounced. An experimental and perhaps theoretical analysis of these cases would certainly be worth pursuing.

In addition, the sample intersection distance metric appears to perform well across many cases, but there remain some open issues. The algorithm as defined simply chooses an input graph from which to draw all edges, and it then checks for their existence in the other graph. The selection of the source graph could have a significant impact on the eventual distance metric, and various adaptions could be explored including randomly choosing a graph for each iteration through the algorithm. Also, the idea of one-way distance is interesting. More specifically, the sample intersection distance metric enables one graph to see another as relatively close, but the second graph does not see the first as close. This defies the identity property of a good distance metric, but it may serve as

the basis for interesting experiments in data mining tasks such as classification and clustering.

Other input types are candidates for sampling for proxies as well. Graphs created from other sorts of data such as citation networks, peer-to-peer networks, networks of linked web pages, or even graphs created from plain text in ways other than word proximity could potentially provide more evidence of the robustness of SLIP as well as open up new application areas. Beyond graphs and sortable permutations, though, the types of inputs which are candidates for SLIP would only seem limited by the types of inputs used throughout computation.

SLIP has been shown to greatly improve efficiency for a variety of computational tasks ranging in inputs from permutations of sortable keys to very large graphs. As the flow of data continues to feed the increased sizes of inputs to practically any problem, techniques such as sampling for large proxy inputs become extremely important to solve problems efficiently when approximations will suffice. Exhaustive processing is not only infeasible but often unnecessary, and SLIP provides a way to reduce the computational load by creating structurally similar proxy inputs whose problem specific features approximate those of the original inputs.

BIBLIOGRAPHY

[1] S.K. Thompson. *Sampling*. John Wiley and Sons, Inc., New York, NY, 1992.

[2] A. Schenker, H. Bunke, M. Last, and A. Kandel. *Graph-Theoretic Techniques for Web Content Mining*. Machine Perception and Artificial Intelligence. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2005.

[3] H.D. Luke. The origins of the sampling theorem. *IEEE Communications Magazine*, 37:106 – 108, 1999.

[4] C.E. Shannon. Communications in the presence of noise. *Proceedings of the IRE*, 237:10–21, 1949.

[5] S.K. Mitra. *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill, 3 edition, 1997.

[6] B. Mitra. *A Course in Digital Signal Processing*. Wiley, 1996.

[7] J. Kurose and K. Ross. *Computer Networking*. Addison-Wesley, Reading, Massachusetts, 2 edition, 2003.

[8] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, B. Schwartz, S.T. Kent, and W. T. Strayer. Single-packet ip traceback. In *IEEE/ACM Transactions on Networking*, volume 10, 2002.

[9] J. M. Gonzalez and V. Paxson. Enhancing network intrusion detection with integrated sampling and filter. In *Proceedings of the 9th International Symposium On Recent Advances In Intrusion Detection (RAID 2006)*, pages 325–336, Hamburg, Germany, September 2006.

[10] N. Duffield, C. Lund, and M. Thorup. Estimating flow distributions from sampled flow statistics. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pages 325–336, New York, NY, USA, 2003. ACM Press.

[11] H. Wang, Y. Lin, Y. Jin, and S.Cheng. Easily-implemented adaptive packet sampling for high speed networks flow measurement. In *Procedings of the Fourth International Conference on Computational Science*, volume 3994, pages 128–135. Springer, 2006.

[12] C. Gkantsidis, M. Mihail, and A. Siberi. Random walks in peer-to-peer networks. *INFOCOM*, 63(3), 2004.

[13] C. Cooper, M. Dyer, and C. Greenhill. Sampling regular graphs and a peer-to-peer network. *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 980–988, 2005.

[14] V. Krishnamurthy, J. Sun, M. Faloutsos, and S. Tauro. Sampling internet topologies: How small can we go? *Proceedings of the International Conference on Internet Computing*, 2003.

[15] V. Krishnamurthy, M. Faloutsos, M. Chrobak, L. Lao, J.H. Cui, and A.G. Percus. Reducing large internet topologies for faster simulations. *Networking*, 2005.

[16] H. Zhang, A. Goel, and R. Govindan. Improving lookup latency in distributed hash table systems using random sampling. *IEEE/ACM Transactions on Networking*, 13(5):1121–1134, 2005.

[17] D. Rafiei and S. Curial. Effectively visualizing large networks through sampling. *Proc. of IEEE Visualization*, pages 375 – 382, 2005.

[18] A.A. Tsay, W.S. Lovejoy, and D.R. Karger. Random sampling in cut, flow, and network design problems. *Mathematics of Operations Research*, 24(2), 1999.

[19] Z. Bar-Yossef and M. Gurevich. Random sampling from a search engine's index. In *Proceedings of the 15th International Conference on World Wide Web*, pages 367–376, New York, NY, 2006. ACM Press.

[20] Z. Bar-Yossef, A. Berg, S. Chien, J. Fakcharoenpohl, and D. Weitz. Approximating aggregate queries about web pages via random walks. In *Proceedings of the International Conference on Very Large Databases*, volume 26, pages 535–544, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[21] U. Srivastava, S. Babu, and J. Widom. Monitoring stream properties for continuous query processing. Technical report, Stanford University, 2003. Retrieved from http://dbpubs.stanford.edu:8090/pub/2003-23 on January 17th, 2005.

[22] S. Chaudhuri, R. Motwani, and V. Narasayya. On random sampling over joins. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, pages 263–274, New York, NY, USA, 1999. ACM Press.

[23] J. F. Naughton and S. Seshadri. On estimating the size of projections. In *ICDT '90: Proceedings of the Third International Conference on Database Theory*, pages 499–513, New York, NY, USA, 1990. Springer-Verlag New York, Inc.

[24] R. J. Lipton, J. F. Naughton, D. A. Schneider, and S. Seshadri. Efficient sampling strategies for relational database operations. In *ICDT: Selected Papers of the 4th International Conference on Database Theory*, pages 195–226, Essex, UK, 1993. Elsevier Science Publishers Ltd.

[25] F. Olken and D. Rotem. Simple random sampling from relational databases. In *VLDB '86: Proceedings of the 12th International Conference on Very Large Data Bases*, pages 160–169, San Francisco, CA, USA, 1986. Morgan Kaufmann Publishers Inc.

[26] F. Olken and D. Rotem. Sampling from spatial databases. In *Proceedings of the Ninth International Conference on Data Engineering*, pages 199–208, Washington, DC, USA, 1993. IEEE Computer Society.

[27] S. Seshadri and J. F. Naughton. Sampling issues in parallel database systems. In *EDBT '92: Proceedings of the 3rd International Conference on Extending Database Technology*, pages 328–343, London, UK, 1992. Springer-Verlag.

[28] S. Brin and L. Page. Dynamic data mining: Exploring large rule spaces by sampling. Technical report, Stanford University, 2005. Retrieved February 5th, 2005 from http://dbpubs.stanford.edu:8090/pub/1999-68.

[29] H. Schutze and C. Silverstein. Projections for efficient document clustering. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 74–81, 1997.

[30] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 631–636, 2006.

[31] D. Stutzbach, R. Rejaie, N. Duffield, S. Sen, and W. Willinger. Sampling techniques for large, dynamics graphs. Technical report, University of Oregon, 2006. In CIS-TR-06-01.

[32] L. Lovasz. Random walks on graphs: A survey. *Combinatorics: Paul Erdos is Eighty*, 2:1–46, 1993.

[33] S.S. Bansal. Sorting using presortedness of input sequence. Technical report, B. Tech., 2002. Retrieved September 20th, 2004 from http://www.cse.iitk.ac.in/research/btp2002/98357.html.

[34] W.H. Burge. Sorting, trees, and measures of order. *Information and Control*, 1(3):181–197, September 1958.

[35] V. Estivill-Castro and D. Wood. A survey of adaptive sorting algorithms. *ACM Computing Survey*, 24(4):441–476, 1992.

[36] H. Manilla. Measures of presortedness and optimal sorting algorithms. *IEEE Transactions on Computers*, 34(3):318–325, 1985.

[37] H. Guo. *Algorithm Selection for Sorting and Probabilistic Inference: A Machine-Learning Approach*. PhD thesis, Kansas State University, 2003.

[38] D. Knuth. *The Art of Computer Programming: Sorting and Searching*, volume 3. Addison-Wesley, Reading, Massachusetts, 2 edition, 1998.

[39] R.A. Wagner and M.J. Fischer. The string-to-string correction problem. *Journal of the Association for Computing Machinery*, 21:168–173, 1974.

[40] K.-C. Tai. The tree-to-tree correction problem. *Journal of the Association for Computing Machinery*, 26(3):422–433, 1979.

[41] T. Eiter and H. Manilla. Distance measures for point sets and their computation. *Acta Informatica*, 34(2):109–133, 1997.

[42] J. Ramon and M. Bruynooghe. A polynomial time computable metric between point sets. *Acta Informatica*, 37(10):765–780, 2001.

[43] R.C. Wilson and E.R. Hancock. Graph matching with hierarchical discrete relaxation. *Pattern Recognition Letters*, 20:1041–1052, 1999.

[44] H. Bunke. Error correcting graph matching: On the influence of the underlying cost function. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):917–922, 1999.

[45] G. Chartrand, G. Kubicki, and M. Shultz. Graph similarity and distance in graphs. *Aequationes Mathematicae*, 55:129–145, 1999.

[46] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(9):689–694, 1997.

[47] H. Bunke and K. Shearer. A graph distance metric on the maximal common subgraph. *Pattern Recognition Letters*, 19:255–259, 1998.

[48] H. Bunke, X. Jiang, and A. Kandel. On the minimum common supergraph of two graphs. *Computing*, 65:13–25, 2000.

[49] W.D. Wallis, P. Shoubridge, M. Kraetz, and D. Ray. Graph distances using graph union. *Pattern Recognition Letters*, 22:701–704, 2001.

[50] M.-L. Fernandez and G. Valiente. A graph distance metric combining maximum common subgraph and minimum common supergraph. *Pattern Recognition*, 22:753–758, 2001.

[51] C.R. Palmer, P.B. Gibbons, and C. Faloutsos. Anf: a fast and scalable tool for data mining in massive graphs. *Proceedings of the Eighth International Conference on Knowledge Discovery and Data Mining*, 2002.

[52] R. Rice. The algorithm selection problem. *Advances in Computers*, 15:65–118, 1976.

[53] R. Sedgewick. Analysis of shellsort and related algorithms. In *ESA '96: Proceedings of the Fourth Annual European Symposium on Algorithms*, pages 1–11, London, UK, 1996. Springer-Verlag.

[54] S. Hettich and S.D. Bay. The uci kdd archive [http://kdd.ics.uci.edu], 1999. Last verified January 24, 2007.

[55] S. Dumais, T. Furnas, T. Landaur, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[56] J.R. Ullman. An algorithm for computing subgraph isomorphism. *Journal of the Association for Computing Machinery*, 23:31–42, 1976.

[57] V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics-Doklady*, 10:707–710, 1966.

[58] C. Bennett and W. D. Potter. Input sampling and sorting algorithm selection. *CSDA Journal on Model Selection (to appear)*, 2005.

[59] M. Kantardzic. *Data Mining: Concepts, Models, Methods, and Algorithms*. IEEE Press, 2003.

[60] A. Freitas. *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer, New York, New York, 2002.

[61] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[62] Yahoo! News. Yahoo! news [http://news.yahoo.com], 2007. Last verified March 1, 2007.

[63] A. Likas, N. Vlassis, and J.J. Verbeek. A global k-means algorithm. *Pattern Recognition*, 36:451–461, 2003.

[64] W.M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.

[65] J. Dunn. Well separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4:95–104, 1974.

Appendix A

Newsgroups Graph Distance Data

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.8656	0.9129	0.9241	0.9298	0.9204	0.9501	0.9486	0.9473	0.9560	0.9529	0.9334	0.9428	0.9372	0.9417	0.9422	0.9461	0.9606	0.9401	0.9504
Religion	0.8656	0.0000	0.9090	0.9062	0.9242	0.8845	0.9504	0.9464	0.9496	0.9546	0.9554	0.9347	0.9440	0.9355	0.9344	0.9431	0.9451	0.9586	0.9424	0.9524
Christian	0.9129	0.9090	0.0000	0.9281	0.9317	0.9218	0.9491	0.9490	0.9513	0.9578	0.9530	0.9378	0.9453	0.9355	0.9429	0.9475	0.9510	0.9608	0.9445	0.9539
Guns	0.9241	0.9062	0.9281	0.0000	0.9185	0.8856	0.9438	0.9443	0.9458	0.9528	0.9509	0.9192	0.9392	0.9287	0.9289	0.9343	0.9381	0.9544	0.9362	0.9446
Mideast	0.9298	0.9242	0.9317	0.9185	0.0000	0.8908	0.9565	0.9552	0.9533	0.9617	0.9590	0.9336	0.9509	0.9400	0.9373	0.9476	0.9510	0.9638	0.9478	0.9514
Misc Politics	0.9204	0.8845	0.9218	0.8856	0.8908	0.0000	0.9447	0.9433	0.9428	0.9511	0.9516	0.9192	0.9388	0.9231	0.9267	0.9355	0.9414	0.9534	0.9363	0.9427
Graphics	0.9501	0.9504	0.9491	0.9438	0.9565	0.9447	0.0000	0.9237	0.9264	0.9357	0.9193	0.9326	0.9298	0.9394	0.9341	0.9435	0.9500	0.9411	0.9476	0.9521
IBM Hardware	0.9486	0.9464	0.9490	0.9443	0.9552	0.9433	0.9237	0.0000	0.9023	0.9317	0.9328	0.9365	0.9203	0.9414	0.9412	0.9356	0.9420	0.9299	0.9412	0.9479
Mac Hardware	0.9473	0.9496	0.9513	0.9458	0.9533	0.9428	0.9264	0.9023	0.0000	0.9390	0.9339	0.9387	0.9199	0.9413	0.9394	0.9338	0.9423	0.9297	0.9404	0.9501
Windows Misc	0.9560	0.9546	0.9578	0.9528	0.9617	0.9511	0.9357	0.9317	0.9390	0.0000	0.9394	0.9460	0.9405	0.9504	0.9507	0.9506	0.9535	0.9478	0.9540	0.9610
Windows X	0.9529	0.9554	0.9530	0.9509	0.9590	0.9516	0.9193	0.9328	0.9339	0.9394	0.0000	0.9391	0.9369	0.9462	0.9410	0.9487	0.9519	0.9465	0.9524	0.9565
Cryptography	0.9334	0.9347	0.9378	0.9192	0.9336	0.9192	0.9326	0.9365	0.9387	0.9460	0.9391	0.0000	0.9323	0.9322	0.9311	0.9398	0.9427	0.9511	0.9393	0.9505
Electronics	0.9428	0.9440	0.9453	0.9392	0.9509	0.9388	0.9298	0.9203	0.9199	0.9405	0.9369	0.9323	0.0000	0.9361	0.9334	0.9304	0.9355	0.9319	0.9392	0.9468
Medicine	0.9372	0.9355	0.9355	0.9287	0.9400	0.9231	0.9394	0.9414	0.9413	0.9504	0.9462	0.9322	0.9361	0.0000	0.9296	0.9403	0.9440	0.9522	0.9427	0.9468
Space	0.9417	0.9344	0.9429	0.9289	0.9373	0.9267	0.9341	0.9412	0.9394	0.9507	0.9410	0.9311	0.9334	0.9296	0.0000	0.9402	0.9460	0.9486	0.9440	0.9488
Autos	0.9422	0.9431	0.9475	0.9343	0.9476	0.9355	0.9435	0.9356	0.9338	0.9506	0.9487	0.9398	0.9304	0.9403	0.9402	0.0000	0.9246	0.9400	0.9352	0.9463
Motorcycles	0.9461	0.9451	0.9510	0.9381	0.9510	0.9414	0.9500	0.9420	0.9423	0.9535	0.9519	0.9427	0.9355	0.9440	0.9460	0.9246	0.0000	0.9478	0.9397	0.9522
For Sale	0.9606	0.9586	0.9608	0.9544	0.9638	0.9534	0.9411	0.9299	0.9297	0.9478	0.9465	0.9511	0.9319	0.9522	0.9486	0.9400	0.9478	0.0000	0.9501	0.9522
Baseball	0.9401	0.9424	0.9445	0.9362	0.9478	0.9363	0.9476	0.9412	0.9404	0.9540	0.9524	0.9393	0.9392	0.9427	0.9440	0.9352	0.9397	0.9501	0.0000	0.9216
Hockey	0.9504	0.9524	0.9539	0.9446	0.9514	0.9427	0.9521	0.9479	0.9501	0.9610	0.9565	0.9505	0.9468	0.9468	0.9488	0.9463	0.9522	0.9522	0.9216	0.0000

Table A.1: MCS Distance between Node-Edge Samples (15%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.8674	0.9157	0.9263	0.9321	0.9231	0.9514	0.9499	0.9486	0.9569	0.9541	0.9354	0.9441	0.9391	0.9436	0.9436	0.9475	0.9615	0.9416	0.9515
Religion	0.8674	0.0000	0.9118	0.9085	0.9265	0.8862	0.9519	0.9477	0.9508	0.9555	0.9566	0.9365	0.9454	0.9373	0.9362	0.9447	0.9466	0.9597	0.9438	0.9536
Christian	0.9157	0.9118	0.0000	0.9303	0.9345	0.9245	0.9506	0.9503	0.9526	0.9588	0.9541	0.9397	0.9468	0.9377	0.9448	0.9489	0.9523	0.9618	0.9459	0.9551
Guns	0.9263	0.9085	0.9303	0.0000	0.9209	0.8880	0.9456	0.9457	0.9471	0.9538	0.9522	0.9216	0.9409	0.9307	0.9310	0.9362	0.9399	0.9556	0.9379	0.9460
Mideast	0.9321	0.9265	0.9345	0.9209	0.0000	0.8937	0.9577	0.9565	0.9545	0.9625	0.9600	0.9357	0.9522	0.9419	0.9394	0.9491	0.9525	0.9648	0.9492	0.9527
Misc Politics	0.9231	0.8862	0.9245	0.8880	0.8937	0.0000	0.9464	0.9448	0.9441	0.9521	0.9529	0.9217	0.9405	0.9254	0.9292	0.9372	0.9430	0.9546	0.9379	0.9442
Graphics	0.9514	0.9519	0.9506	0.9456	0.9577	0.9464	0.0000	0.9256	0.9284	0.9370	0.9213	0.9347	0.9317	0.9411	0.9362	0.9451	0.9514	0.9425	0.9488	0.9532
IBM Hardware	0.9499	0.9477	0.9503	0.9457	0.9565	0.9448	0.9256	0.0000	0.9046	0.9329	0.9344	0.9383	0.9221	0.9428	0.9429	0.9374	0.9434	0.9315	0.9424	0.9491
Mac Hardware	0.9486	0.9508	0.9526	0.9471	0.9545	0.9441	0.9284	0.9046	0.0000	0.9403	0.9354	0.9403	0.9219	0.9429	0.9410	0.9355	0.9438	0.9316	0.9417	0.9513
Windows Misc	0.9569	0.9555	0.9588	0.9538	0.9625	0.9521	0.9370	0.9329	0.9403	0.0000	0.9406	0.9472	0.9418	0.9514	0.9518	0.9517	0.9544	0.9487	0.9549	0.9616
Windows X	0.9541	0.9566	0.9541	0.9522	0.9600	0.9529	0.9213	0.9344	0.9354	0.9406	0.0000	0.9406	0.9383	0.9476	0.9426	0.9498	0.9530	0.9475	0.9535	0.9574
Cryptography	0.9354	0.9365	0.9397	0.9216	0.9357	0.9217	0.9347	0.9383	0.9403	0.9472	0.9406	0.0000	0.9343	0.9344	0.9333	0.9416	0.9442	0.9523	0.9407	0.9517
Electronics	0.9441	0.9454	0.9468	0.9409	0.9522	0.9405	0.9317	0.9221	0.9219	0.9418	0.9383	0.9343	0.0000	0.9378	0.9353	0.9322	0.9371	0.9335	0.9406	0.9480
Medicine	0.9391	0.9373	0.9377	0.9307	0.9419	0.9254	0.9411	0.9428	0.9429	0.9514	0.9476	0.9344	0.9378	0.0000	0.9319	0.9420	0.9456	0.9534	0.9443	0.9482
Space	0.9436	0.9362	0.9448	0.9310	0.9394	0.9292	0.9362	0.9429	0.9410	0.9518	0.9426	0.9333	0.9353	0.9319	0.0000	0.9420	0.9476	0.9501	0.9455	0.9501
Autos	0.9436	0.9447	0.9489	0.9362	0.9491	0.9372	0.9451	0.9374	0.9355	0.9517	0.9498	0.9416	0.9322	0.9420	0.9420	0.0000	0.9268	0.9415	0.9366	0.9475
Motorcycles	0.9475	0.9466	0.9523	0.9399	0.9525	0.9430	0.9514	0.9434	0.9438	0.9544	0.9530	0.9442	0.9371	0.9456	0.9476	0.9268	0.0000	0.9491	0.9413	0.9534
For Sale	0.9615	0.9597	0.9618	0.9556	0.9648	0.9546	0.9425	0.9315	0.9316	0.9487	0.9475	0.9523	0.9335	0.9534	0.9501	0.9415	0.9491	0.0000	0.9513	0.9533
Baseball	0.9416	0.9438	0.9459	0.9379	0.9492	0.9379	0.9488	0.9424	0.9417	0.9549	0.9535	0.9407	0.9406	0.9443	0.9455	0.9366	0.9413	0.9513	0.0000	0.9232
Hockey	0.9515	0.9536	0.9551	0.9460	0.9527	0.9442	0.9532	0.9491	0.9513	0.9616	0.9574	0.9517	0.9480	0.9482	0.9501	0.9475	0.9534	0.9533	0.9232	0.0000

Table A.2: WGU Distance between Node-Edge Samples (15%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0	8897	9614	9753	11038	10584	10139	9056	8752	12209	11079	9969	9084	10361	10444	9172	8899	9334	8991	9957
Religion	8897	0	10013	9880	11403	10275	10662	9525	9303	12692	11652	10490	9615	10822	10787	9697	9390	9823	9536	10514
Christian	9614	10013	0	10121	11382	10900	10423	9364	9124	12561	11383	10351	9432	10625	10766	9570	9287	9648	9367	10333
Guns	9753	9880	10121	0	11007	10063	10248	9207	8953	12364	11266	9914	9251	10408	10403	9259	8988	9457	9148	10076
Mideast	11038	11403	11382	11007	0	11262	11733	10638	10312	13827	12683	11377	10682	11839	11768	10712	10437	10882	10569	11431
Misc Politics	10584	10275	10900	10063	11262	0	11201	10124	9832	13269	12231	10807	10170	11197	11270	10198	9975	10390	10069	10973
Graphics	10139	10662	10423	10248	11733	11201	0	8731	8511	11840	10478	10064	8969	10512	10397	9309	9080	9095	9234	10104
IBM Hardware	9056	9525	9364	9207	10638	10124	8731	0	7152	10725	9737	9109	7802	9511	9502	8140	7897	7882	8079	8971
Mac Hardware	8752	9303	9124	8953	10312	9832	8511	7152	0	10609	9483	8869	7532	9235	9188	7836	7625	7612	7791	8727
Windows Misc	12209	12692	12561	12364	13827	13269	11840	10725	10609	0	12786	12240	11051	12668	12667	11371	11082	11135	11296	12252
Windows X	11079	11652	11383	11266	12683	12231	10478	9737	9483	12786	0	11046	9945	11524	11397	10275	9992	10061	10204	11080
Cryptography	9969	10490	10351	9914	11377	10807	10064	9109	8869	12240	11046	0	9165	10522	10487	9397	9104	9433	9238	10230
Electronics	9084	9615	9432	9251	10682	10170	8969	7802	7532	11051	9945	9165	0	9547	9488	8186	7929	8044	8181	9085
Medicine	10361	10822	10625	10408	11839	11197	10512	9511	9235	12668	11524	10522	9547	0	10771	9723	9450	9779	9624	10480
Space	10444	10787	10766	10403	11768	11270	10397	9502	9188	12667	11397	10487	9488	10771	0	9712	9477	9702	9637	10509
Autos	9172	9697	9570	9259	10712	10198	9309	8140	7836	11371	10275	9397	8186	9723	9712	0	7859	8270	8211	9173
Motorcycles	8899	9390	9287	8988	10437	9975	9080	7897	7625	11082	9992	9104	7929	9450	9477	7859	0	8051	7948	8932
For Sale	9334	9823	9648	9457	10882	10390	9095	7882	7612	11135	10061	9433	8044	9779	9702	8270	8051	0	8289	9111
Baseball	8991	9536	9367	9148	10569	10069	9234	8079	7791	11296	10204	9238	8181	9624	9637	8211	7948	8289	0	8598
Hockey	9957	10514	10333	10076	11431	10973	10104	8971	8727	12252	11080	10230	9085	10480	10509	9173	8932	9111	8598	0

Table A.3: UGU Distance between Node-Edge Samples (15%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.5437	0.7938	0.7409	0.7599	0.7278	0.7673	0.7540	0.7690	0.7590	0.7677	0.7616	0.7620	0.7501	0.7600	0.7515	0.7629	0.7812	0.7604	0.7726
Religion	0.5437	0.0000	0.7846	0.7023	0.7475	0.6386	0.7601	0.7580	0.7612	0.7514	0.7654	0.7614	0.7547	0.7428	0.7568	0.7532	0.7587	0.7719	0.7647	0.7757
Christian	0.7938	0.7846	0.0000	0.7935	0.8036	0.8001	0.8064	0.8135	0.8139	0.8044	0.8055	0.8054	0.8049	0.7946	0.8007	0.8067	0.8126	0.8215	0.8128	0.8198
Guns	0.7409	0.7023	0.7935	0.0000	0.7301	0.6379	0.7417	0.7518	0.7467	0.7329	0.7426	0.7127	0.7370	0.7216	0.7387	0.7275	0.7343	0.7533	0.7392	0.7481
Mideast	0.7599	0.7475	0.8036	0.7301	0.0000	0.7409	0.8361	0.8417	0.8370	0.8404	0.8355	0.8329	0.8373	0.8258	0.8236	0.8368	0.8428	0.8463	0.8370	0.8406
Misc Politics	0.7278	0.6386	0.8001	0.6379	0.7409	0.0000	0.7795	0.7795	0.7860	0.7714	0.7859	0.7604	0.7754	0.7548	0.7668	0.7683	0.7783	0.7892	0.7803	0.7839
Graphics	0.7673	0.7601	0.8064	0.7417	0.8361	0.7795	0.0000	0.7250	0.7489	0.7070	0.6946	0.7427	0.7399	0.7482	0.7395	0.7604	0.7661	0.7507	0.7655	0.7699
IBM Hardware	0.7540	0.7580	0.8135	0.7518	0.8417	0.7795	0.7250	0.0000	0.6463	0.6301	0.6691	0.7023	0.6563	0.6897	0.7021	0.6940	0.7048	0.6535	0.7043	0.7033
Mac Hardware	0.7690	0.7612	0.8139	0.7467	0.8370	0.7860	0.7489	0.6463	0.0000	0.6513	0.6674	0.6911	0.6573	0.6710	0.6824	0.6707	0.6930	0.6642	0.6850	0.6970
Windows Misc	0.7590	0.7514	0.8044	0.7329	0.8404	0.7714	0.7070	0.6301	0.6513	0.0000	0.6853	0.7321	0.7137	0.7253	0.7366	0.7310	0.7370	0.7241	0.7406	0.7505
Windows X	0.7677	0.7654	0.8055	0.7426	0.8355	0.7859	0.6946	0.6691	0.6674	0.6853	0.0000	0.7752	0.7762	0.7955	0.7821	0.7929	0.8039	0.7853	0.8062	0.8091
Cryptography	0.7616	0.7614	0.8054	0.7127	0.8329	0.7604	0.7427	0.7023	0.6911	0.7321	0.7752	0.0000	0.7791	0.7776	0.7771	0.7882	0.7886	0.8000	0.7882	0.8010
Electronics	0.7620	0.7547	0.8049	0.7370	0.8373	0.7754	0.7399	0.6563	0.6573	0.7137	0.7762	0.7791	0.0000	0.6433	0.6569	0.6571	0.6676	0.6646	0.6735	0.6751
Medicine	0.7501	0.7428	0.7946	0.7216	0.8258	0.7548	0.7482	0.6897	0.6710	0.7253	0.7955	0.7776	0.6433	0.0000	0.7409	0.7581	0.7595	0.7655	0.7555	0.7584
Space	0.7600	0.7568	0.8007	0.7387	0.8236	0.7668	0.7395	0.7021	0.6824	0.7366	0.7821	0.7771	0.6569	0.7409	0.0000	0.7792	0.7906	0.7903	0.7848	0.7961
Autos	0.7515	0.7532	0.8067	0.7275	0.8368	0.7683	0.7604	0.6940	0.6707	0.7310	0.7929	0.7882	0.6571	0.7581	0.7792	0.0000	0.6672	0.6818	0.6682	0.6864
Motorcycles	0.7629	0.7587	0.8126	0.7343	0.8428	0.7783	0.7661	0.7048	0.6930	0.7370	0.8039	0.7886	0.6676	0.7595	0.7906	0.6672	0.0000	0.7074	0.6938	0.7011
For Sale	0.7812	0.7719	0.8215	0.7533	0.8463	0.7892	0.7507	0.6535	0.6642	0.7241	0.7853	0.8000	0.6646	0.7655	0.7903	0.6818	0.7074	0.0000	0.7238	0.7260
Baseball	0.7604	0.7647	0.8128	0.7392	0.8370	0.7803	0.7655	0.7043	0.6850	0.7406	0.8062	0.7882	0.6735	0.7555	0.7848	0.6682	0.6938	0.7238	0.0000	0.6710
Hockey	0.7726	0.7757	0.8198	0.7481	0.8406	0.7839	0.7699	0.7033	0.6970	0.7505	0.8091	0.8010	0.6751	0.7584	0.7961	0.6864	0.7011	0.7260	0.6710	0.0000

Table A.4: SID Distance between Node-Edge Samples (15%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.9476	0.9813	0.9844	0.9854	0.9812	0.9853	0.9806	0.9850	0.9877	0.9861	0.9835	0.9774	0.9794	0.9840	0.9754	0.9818	0.9833	0.9864	0.9866
Religion	0.9476	0.0000	0.9729	0.9805	0.9803	0.9634	0.9816	0.9778	0.9817	0.9845	0.9854	0.9837	0.9744	0.9752	0.9829	0.9702	0.9835	0.9788	0.9841	0.9874
Christian	0.9813	0.9729	0.0000	0.9865	0.9835	0.9861	0.9818	0.9834	0.9840	0.9848	0.9890	0.9865	0.9790	0.9800	0.9877	0.9780	0.9897	0.9844	0.9853	0.9902
Guns	0.9844	0.9805	0.9865	0.0000	0.9861	0.9782	0.9869	0.9860	0.9832	0.9839	0.9891	0.9816	0.9855	0.9816	0.9893	0.9767	0.9851	0.9876	0.9893	0.9902
Mideast	0.9854	0.9803	0.9835	0.9861	0.0000	0.9770	0.9864	0.9852	0.9893	0.9892	0.9897	0.9890	0.9839	0.9839	0.9893	0.9802	0.9899	0.9872	0.9898	0.9929
Misc Politics	0.9812	0.9634	0.9861	0.9782	0.9770	0.0000	0.9859	0.9841	0.9844	0.9860	0.9891	0.9840	0.9826	0.9798	0.9843	0.9753	0.9819	0.9852	0.9856	0.9862
Graphics	0.9853	0.9816	0.9818	0.9869	0.9864	0.9859	0.0000	0.9794	0.9846	0.9774	0.9822	0.9848	0.9717	0.9782	0.9866	0.9742	0.9843	0.9785	0.9852	0.9871
IBM Hardware	0.9806	0.9778	0.9834	0.9860	0.9852	0.9841	0.9794	0.0000	0.9779	0.9739	0.9801	0.9829	0.9639	0.9753	0.9828	0.9708	0.9823	0.9756	0.9812	0.9846
Mac Hardware	0.9850	0.9817	0.9840	0.9832	0.9893	0.9844	0.9846	0.9779	0.0000	0.9792	0.9881	0.9833	0.9772	0.9788	0.9880	0.9716	0.9841	0.9779	0.9871	0.9857
Windows Misc	0.9877	0.9845	0.9848	0.9839	0.9892	0.9860	0.9774	0.9739	0.9792	0.0000	0.9798	0.9850	0.9813	0.9815	0.9876	0.9764	0.9857	0.9816	0.9851	0.9888
Windows X	0.9861	0.9854	0.9890	0.9891	0.9897	0.9891	0.9822	0.9801	0.9881	0.9798	0.0000	0.9858	0.9811	0.9832	0.9864	0.9812	0.9856	0.9858	0.9892	0.9908
Cryptography	0.9835	0.9837	0.9865	0.9816	0.9890	0.9840	0.9848	0.9829	0.9833	0.9850	0.9858	0.0000	0.9828	0.9810	0.9845	0.9778	0.9836	0.9839	0.9865	0.9888
Electronics	0.9774	0.9744	0.9790	0.9855	0.9839	0.9826	0.9717	0.9639	0.9772	0.9813	0.9811	0.9828	0.0000	0.9643	0.9796	0.9631	0.9775	0.9705	0.9823	0.9829
Medicine	0.9794	0.9752	0.9800	0.9816	0.9839	0.9798	0.9782	0.9753	0.9788	0.9815	0.9832	0.9810	0.9643	0.0000	0.9800	0.9684	0.9759	0.9785	0.9857	0.9843
Space	0.9840	0.9829	0.9877	0.9893	0.9893	0.9843	0.9866	0.9828	0.9880	0.9876	0.9864	0.9845	0.9796	0.9800	0.0000	0.9789	0.9826	0.9863	0.9854	0.9921
Autos	0.9754	0.9702	0.9780	0.9767	0.9802	0.9753	0.9742	0.9708	0.9716	0.9764	0.9812	0.9778	0.9631	0.9684	0.9789	0.0000	0.9709	0.9697	0.9761	0.9818
Motorcycles	0.9818	0.9835	0.9897	0.9851	0.9899	0.9819	0.9843	0.9823	0.9841	0.9857	0.9856	0.9836	0.9775	0.9759	0.9826	0.9709	0.0000	0.9806	0.9864	0.9835
For Sale	0.9833	0.9788	0.9844	0.9876	0.9872	0.9852	0.9785	0.9756	0.9779	0.9816	0.9858	0.9839	0.9705	0.9785	0.9863	0.9697	0.9806	0.0000	0.9817	0.9832
Baseball	0.9864	0.9841	0.9853	0.9893	0.9898	0.9856	0.9852	0.9812	0.9871	0.9851	0.9892	0.9865	0.9823	0.9857	0.9854	0.9761	0.9864	0.9817	0.0000	0.9819
Hockey	0.9866	0.9874	0.9902	0.9902	0.9929	0.9862	0.9871	0.9846	0.9857	0.9888	0.9908	0.9888	0.9829	0.9843	0.9921	0.9818	0.9835	0.9832	0.9819	0.0000

Table A.5: MCS Distance between Random Walk Samples (15%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.9517	0.9827	0.9853	0.9863	0.9824	0.9859	0.9814	0.9856	0.9882	0.9867	0.9843	0.9784	0.9805	0.9848	0.9765	0.9826	0.9839	0.9870	0.9872
Religion	0.9517	0.0000	0.9750	0.9818	0.9817	0.9662	0.9825	0.9788	0.9825	0.9852	0.9861	0.9847	0.9756	0.9766	0.9838	0.9715	0.9843	0.9796	0.9849	0.9880
Christian	0.9827	0.9750	0.0000	0.9874	0.9847	0.9873	0.9828	0.9841	0.9846	0.9855	0.9896	0.9873	0.9801	0.9814	0.9885	0.9791	0.9902	0.9851	0.9861	0.9907
Guns	0.9853	0.9818	0.9874	0.0000	0.9871	0.9800	0.9876	0.9867	0.9839	0.9846	0.9896	0.9827	0.9863	0.9828	0.9899	0.9779	0.9859	0.9882	0.9899	0.9906
Mideast	0.9863	0.9817	0.9847	0.9871	0.0000	0.9791	0.9871	0.9858	0.9897	0.9898	0.9902	0.9897	0.9847	0.9850	0.9900	0.9812	0.9904	0.9877	0.9903	0.9933
Misc Politics	0.9824	0.9662	0.9873	0.9800	0.9791	0.0000	0.9868	0.9849	0.9850	0.9867	0.9897	0.9852	0.9836	0.9813	0.9854	0.9767	0.9828	0.9858	0.9864	0.9870
Graphics	0.9859	0.9825	0.9828	0.9876	0.9871	0.9868	0.0000	0.9807	0.9854	0.9789	0.9834	0.9858	0.9733	0.9796	0.9875	0.9755	0.9851	0.9796	0.9859	0.9877
IBM Hardware	0.9814	0.9788	0.9841	0.9867	0.9858	0.9849	0.9807	0.0000	0.9791	0.9754	0.9811	0.9838	0.9660	0.9765	0.9837	0.9723	0.9831	0.9769	0.9820	0.9852
Mac Hardware	0.9856	0.9825	0.9846	0.9839	0.9897	0.9850	0.9854	0.9791	0.0000	0.9801	0.9887	0.9840	0.9784	0.9798	0.9886	0.9728	0.9848	0.9788	0.9876	0.9862
Windows Misc	0.9882	0.9852	0.9855	0.9846	0.9898	0.9867	0.9789	0.9754	0.9801	0.0000	0.9810	0.9858	0.9824	0.9824	0.9882	0.9775	0.9863	0.9823	0.9857	0.9893
Windows X	0.9867	0.9861	0.9896	0.9896	0.9902	0.9897	0.9834	0.9811	0.9887	0.9810	0.0000	0.9866	0.9821	0.9841	0.9872	0.9821	0.9861	0.9865	0.9896	0.9912
Cryptography	0.9843	0.9847	0.9873	0.9827	0.9897	0.9852	0.9858	0.9838	0.9840	0.9858	0.9866	0.0000	0.9839	0.9822	0.9854	0.9789	0.9843	0.9846	0.9871	0.9892
Electronics	0.9784	0.9756	0.9801	0.9863	0.9847	0.9836	0.9733	0.9660	0.9784	0.9824	0.9821	0.9839	0.0000	0.9662	0.9808	0.9651	0.9786	0.9720	0.9832	0.9838
Medicine	0.9805	0.9766	0.9814	0.9828	0.9850	0.9813	0.9796	0.9765	0.9798	0.9824	0.9841	0.9822	0.9662	0.0000	0.9813	0.9700	0.9770	0.9795	0.9864	0.9851
Space	0.9848	0.9838	0.9885	0.9899	0.9900	0.9854	0.9875	0.9837	0.9886	0.9882	0.9872	0.9854	0.9808	0.9813	0.0000	0.9800	0.9834	0.9870	0.9860	0.9925
Autos	0.9765	0.9715	0.9791	0.9779	0.9812	0.9767	0.9755	0.9723	0.9728	0.9775	0.9821	0.9789	0.9651	0.9700	0.9800	0.0000	0.9724	0.9710	0.9774	0.9827
Motorcycles	0.9826	0.9843	0.9902	0.9859	0.9904	0.9828	0.9851	0.9831	0.9848	0.9863	0.9861	0.9843	0.9786	0.9770	0.9834	0.9724	0.0000	0.9814	0.9871	0.9841
For Sale	0.9839	0.9796	0.9851	0.9882	0.9877	0.9858	0.9796	0.9769	0.9788	0.9823	0.9865	0.9846	0.9720	0.9795	0.9870	0.9710	0.9814	0.0000	0.9825	0.9838
Baseball	0.9870	0.9849	0.9861	0.9899	0.9903	0.9864	0.9859	0.9820	0.9876	0.9857	0.9896	0.9871	0.9832	0.9864	0.9860	0.9774	0.9871	0.9825	0.0000	0.9830
Hockey	0.9872	0.9880	0.9907	0.9906	0.9933	0.9870	0.9877	0.9852	0.9862	0.9893	0.9912	0.9892	0.9838	0.9851	0.9925	0.9827	0.9841	0.9838	0.9830	0.0000

Table A.6: WGU Distance between Random Walk Samples (15%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0	10547	10995	10977	12304	11922	10865	9646	9423	12998	11826	10997	9729	11257	11342	9795	9547	9762	9847	10694
Religion	10547	0	11366	11446	12735	12069	11336	10137	9910	13469	12361	11552	10214	11708	11867	10232	10126	10221	10354	11263
Christian	10995	11366	0	11346	12587	12361	11116	10019	9726	13249	12217	11386	10080	11596	11749	10164	10018	10107	10152	11095
Guns	10977	11446	11346	0	12571	12107	11148	9995	9636	13149	12141	11206	10130	11554	11707	10066	9856	10093	10152	11017
Mideast	12304	12735	12587	12571	0	13370	12445	11282	11063	14600	13472	12677	11399	12905	13022	11423	11261	11392	11477	12398
Misc Politics	11922	12069	12361	12107	13370	0	12145	10972	10671	14220	13166	12273	11085	12523	12614	11037	10803	11060	11097	11952
Graphics	10865	11336	11116	11148	12445	12145	0	9751	9540	12875	11867	11148	9748	11354	11523	9894	9714	9797	9946	10825
IBM Hardware	9646	10137	10019	9995	11282	10972	9751	0	8305	11678	10692	9979	8519	10175	10310	8729	8551	8634	8747	9642
Mac Hardware	9423	9910	9726	9636	11063	10671	9540	8305	0	11489	10549	9680	8434	9942	10105	8442	8276	8367	8540	9357
Windows Misc	12998	13469	13249	13149	14600	14220	12875	11678	11489	0	13862	13221	11985	13479	13624	11971	11813	11910	12015	12948
Windows X	11826	12361	12217	12141	13472	13166	11867	10692	10549	13862	0	12111	10855	12397	12460	10959	10677	10876	10969	11854
Cryptography	10997	11552	11386	11206	12677	12273	11148	9979	9680	13221	12111	0	10122	11580	11643	10126	9866	10063	10138	11027
Electronics	9729	10214	10080	10130	11399	11085	9748	8519	8434	11985	10855	10122	0	10106	10393	8742	8616	8689	8910	9757
Medicine	11257	11708	11596	11554	12905	12523	11354	10175	9942	13479	12397	11580	10106	0	11891	10284	10064	10303	10472	11283
Space	11342	11867	11749	11707	13022	12614	11523	10310	10105	13624	12460	11643	10393	11891	0	10479	10181	10446	10451	11438
Autos	9795	10232	10164	10066	11423	11037	9894	8729	8442	11971	10959	10126	8742	10284	10479	0	8612	8773	8910	9841
Motorcycles	9547	10126	10018	9856	11261	10803	9714	8551	8276	11813	10677	9866	8616	10064	10181	8612	0	8589	8712	9499
For Sale	9762	10221	10107	10093	11392	11060	9797	8634	8367	11910	10876	10063	8689	10303	10446	8773	8589	0	8823	9684
Baseball	9847	10354	10152	10152	11477	11097	9946	8747	8540	12015	10969	10138	8910	10472	10451	8910	8712	8823	0	9691
Hockey	10694	11263	11095	11017	12398	11952	10825	9642	9357	12948	11854	11027	9757	11283	11438	9841	9499	9684	9691	0

Table A.7: UGU Distance between Random Walk Samples (15%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.7279	0.8766	0.8302	0.9021	0.8145	0.8671	0.8420	0.8619	0.8164	0.8311	0.8217	0.8520	0.8224	0.8532	0.8449	0.8422	0.8344	0.8605	0.8834
Religion	0.7279	0.0000	0.8726	0.8775	0.8704	0.8315	0.8721	0.8534	0.8902	0.8366	0.8570	0.8623	0.8647	0.8591	0.8757	0.8528	0.8864	0.8498	0.9157	0.9400
Christian	0.8766	0.8726	0.0000	0.9065	0.9053	0.8724	0.8955	0.8852	0.8947	0.8334	0.8742	0.8878	0.8726	0.8895	0.8919	0.8856	0.9179	0.8647	0.9035	0.9451
Guns	0.8302	0.8775	0.9065	0.0000	0.8512	0.8173	0.8578	0.8276	0.8713	0.7966	0.8183	0.7974	0.8685	0.8572	0.8574	0.8233	0.8722	0.8650	0.8757	0.8664
Mideast	0.9021	0.8704	0.9053	0.8512	0.0000	0.9190	0.9445	0.9294	0.9688	0.9265	0.9350	0.9352	0.9434	0.9262	0.9405	0.9355	0.9515	0.9477	0.9409	0.9568
Misc Politics	0.8145	0.8315	0.8724	0.8173	0.9190	0.0000	0.8963	0.8844	0.8938	0.8496	0.8585	0.8559	0.8836	0.8568	0.8769	0.8705	0.8855	0.8684	0.8933	0.8752
Graphics	0.8671	0.8721	0.8955	0.8578	0.9445	0.8963	0.0000	0.8831	0.8944	0.8376	0.8638	0.8727	0.8578	0.8537	0.9124	0.8732	0.9045	0.8694	0.8862	0.9067
IBM Hardware	0.8420	0.8534	0.8852	0.8276	0.9294	0.8844	0.8831	0.0000	0.8492	0.7528	0.7914	0.8039	0.7920	0.8112	0.8231	0.7804	0.8251	0.7969	0.8542	0.8597
Mac Hardware	0.8619	0.8902	0.8947	0.8713	0.9688	0.8938	0.8944	0.8492	0.0000	0.7331	0.8258	0.8017	0.8269	0.8164	0.8314	0.7992	0.8741	0.7961	0.8325	0.8569
Windows Misc	0.8164	0.8366	0.8334	0.7966	0.9265	0.8496	0.8376	0.7528	0.7331	0.0000	0.8239	0.8407	0.8778	0.8642	0.8821	0.8470	0.8619	0.8462	0.8744	0.8926
Windows X	0.8311	0.8570	0.8742	0.8183	0.9350	0.8585	0.8638	0.7914	0.8258	0.8239	0.0000	0.8676	0.8861	0.8740	0.8777	0.8719	0.9082	0.8726	0.9190	0.9127
Cryptography	0.8217	0.8623	0.8878	0.7974	0.9352	0.8559	0.8727	0.8039	0.8017	0.8407	0.8676	0.0000	0.9111	0.8871	0.9062	0.8783	0.8961	0.8939	0.9075	0.9073
Electronics	0.8520	0.8647	0.8726	0.8685	0.9434	0.8836	0.8578	0.7920	0.8269	0.8778	0.8861	0.9111	0.0000	0.7328	0.8393	0.7924	0.8344	0.8009	0.8547	0.8253
Medicine	0.8224	0.8591	0.8895	0.8572	0.9262	0.8568	0.8537	0.8112	0.8164	0.8642	0.8740	0.8871	0.7328	0.0000	0.8731	0.8691	0.8733	0.8927	0.9124	0.9243
Space	0.8532	0.8757	0.8919	0.8574	0.9405	0.8769	0.9124	0.8231	0.8314	0.8821	0.8777	0.9062	0.8393	0.8731	0.0000	0.8527	0.8913	0.8682	0.8971	0.8876
Autos	0.8449	0.8528	0.8856	0.8233	0.9355	0.8705	0.8732	0.7804	0.7992	0.8470	0.8719	0.8783	0.7924	0.8691	0.8527	0.0000	0.8089	0.7929	0.8173	0.8300
Motorcycles	0.8422	0.8864	0.9179	0.8722	0.9515	0.8855	0.9045	0.8251	0.8741	0.8619	0.9082	0.8961	0.8344	0.8733	0.8913	0.8089	0.0000	0.8634	0.8656	0.8620
For Sale	0.8344	0.8498	0.8647	0.8650	0.9477	0.8684	0.8694	0.7969	0.7961	0.8462	0.8726	0.8939	0.8009	0.8927	0.8682	0.7929	0.8634	0.0000	0.8664	0.8721
Baseball	0.8605	0.9157	0.9035	0.8757	0.9409	0.8933	0.8862	0.8542	0.8325	0.8744	0.9190	0.9075	0.8547	0.9124	0.8971	0.8173	0.8656	0.8664	0.0000	0.8352
Hockey	0.8834	0.9400	0.9451	0.8664	0.9568	0.8752	0.9067	0.8597	0.8569	0.8926	0.9127	0.9073	0.8253	0.9243	0.8876	0.8300	0.8620	0.8721	0.8352	0.0000

Table A.8: SID Distance between Random Walk Samples (15%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.8442	0.9136	0.9296	0.9363	0.9254	0.9550	0.9525	0.9501	0.9616	0.9592	0.9385	0.9479	0.9446	0.9440	0.9472	0.9510	0.9641	0.9460	0.9550
Religion	0.8442	0.0000	0.9108	0.9001	0.9288	0.8677	0.9538	0.9506	0.9508	0.9592	0.9596	0.9379	0.9474	0.9371	0.9334	0.9469	0.9484	0.9624	0.9457	0.9574
Christian	0.9136	0.9108	0.0000	0.9350	0.9365	0.9280	0.9553	0.9531	0.9528	0.9630	0.9603	0.9435	0.9496	0.9420	0.9470	0.9498	0.9524	0.9630	0.9482	0.9577
Guns	0.9296	0.9001	0.9350	0.0000	0.9181	0.8757	0.9498	0.9467	0.9453	0.9594	0.9567	0.9230	0.9436	0.9303	0.9312	0.9381	0.9415	0.9574	0.9409	0.9501
Mideast	0.9363	0.9288	0.9365	0.9181	0.0000	0.8898	0.9600	0.9624	0.9574	0.9680	0.9654	0.9425	0.9575	0.9431	0.9429	0.9535	0.9563	0.9680	0.9537	0.9575
Misc Politics	0.9254	0.8677	0.9280	0.8757	0.8898	0.0000	0.9525	0.9497	0.9445	0.9588	0.9581	0.9283	0.9447	0.9308	0.9293	0.9400	0.9445	0.9566	0.9415	0.9483
Graphics	0.9550	0.9538	0.9553	0.9498	0.9600	0.9525	0.0000	0.9291	0.9286	0.9386	0.9255	0.9390	0.9365	0.9462	0.9378	0.9486	0.9511	0.9426	0.9526	0.9571
IBM Hardware	0.9525	0.9506	0.9531	0.9467	0.9624	0.9497	0.9291	0.0000	0.9065	0.9318	0.9371	0.9438	0.9228	0.9464	0.9459	0.9381	0.9435	0.9297	0.9455	0.9533
Mac Hardware	0.9501	0.9508	0.9528	0.9453	0.9574	0.9445	0.9286	0.9065	0.0000	0.9413	0.9383	0.9422	0.9232	0.9427	0.9410	0.9365	0.9416	0.9287	0.9426	0.9534
Windows Misc	0.9616	0.9592	0.9630	0.9594	0.9680	0.9588	0.9386	0.9318	0.9413	0.0000	0.9483	0.9537	0.9485	0.9559	0.9563	0.9551	0.9590	0.9522	0.9591	0.9649
Windows X	0.9592	0.9596	0.9603	0.9567	0.9654	0.9581	0.9255	0.9371	0.9383	0.9483	0.0000	0.9460	0.9445	0.9526	0.9502	0.9535	0.9559	0.9529	0.9574	0.9629
Cryptography	0.9385	0.9379	0.9435	0.9230	0.9425	0.9283	0.9390	0.9438	0.9422	0.9537	0.9460	0.0000	0.9382	0.9401	0.9346	0.9445	0.9488	0.9546	0.9492	0.9578
Electronics	0.9479	0.9474	0.9496	0.9436	0.9575	0.9447	0.9365	0.9228	0.9232	0.9485	0.9445	0.9382	0.0000	0.9409	0.9375	0.9306	0.9377	0.9355	0.9442	0.9541
Medicine	0.9446	0.9371	0.9420	0.9303	0.9431	0.9308	0.9462	0.9464	0.9427	0.9559	0.9526	0.9401	0.9409	0.0000	0.9313	0.9428	0.9464	0.9541	0.9459	0.9545
Space	0.9440	0.9334	0.9470	0.9312	0.9429	0.9293	0.9378	0.9459	0.9410	0.9563	0.9502	0.9346	0.9375	0.9313	0.0000	0.9423	0.9486	0.9514	0.9481	0.9548
Autos	0.9472	0.9469	0.9498	0.9381	0.9535	0.9400	0.9486	0.9381	0.9365	0.9551	0.9535	0.9445	0.9306	0.9428	0.9423	0.0000	0.9226	0.9410	0.9400	0.9496
Motorcycles	0.9510	0.9484	0.9524	0.9415	0.9563	0.9445	0.9511	0.9435	0.9416	0.9590	0.9559	0.9488	0.9377	0.9464	0.9486	0.9226	0.0000	0.9511	0.9424	0.9540
For Sale	0.9641	0.9624	0.9630	0.9574	0.9680	0.9566	0.9426	0.9297	0.9287	0.9522	0.9529	0.9546	0.9355	0.9541	0.9514	0.9410	0.9511	0.0000	0.9539	0.9554
Baseball	0.9460	0.9457	0.9482	0.9409	0.9537	0.9415	0.9526	0.9455	0.9426	0.9591	0.9574	0.9492	0.9442	0.9459	0.9481	0.9400	0.9424	0.9539	0.0000	0.9294
Hockey	0.9550	0.9574	0.9577	0.9501	0.9575	0.9483	0.9571	0.9533	0.9534	0.9649	0.9629	0.9578	0.9541	0.9545	0.9548	0.9496	0.9540	0.9554	0.9294	0.0000

Table A.9: MCS Distance between Node Edge Samples (30%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.8490	0.9187	0.9333	0.9398	0.9295	0.9569	0.9543	0.9522	0.9628	0.9608	0.9417	0.9501	0.9475	0.9470	0.9494	0.9532	0.9654	0.9483	0.9567
Religion	0.8490	0.0000	0.9159	0.9041	0.9324	0.8715	0.9558	0.9524	0.9527	0.9604	0.9611	0.9410	0.9495	0.9401	0.9364	0.9492	0.9505	0.9639	0.9480	0.9591
Christian	0.9187	0.9159	0.0000	0.9386	0.9403	0.9322	0.9573	0.9549	0.9548	0.9642	0.9618	0.9465	0.9518	0.9451	0.9499	0.9522	0.9546	0.9644	0.9505	0.9594
Guns	0.9333	0.9041	0.9386	0.0000	0.9222	0.8801	0.9520	0.9489	0.9476	0.9608	0.9584	0.9267	0.9462	0.9336	0.9345	0.9410	0.9442	0.9592	0.9435	0.9522
Mideast	0.9398	0.9324	0.9403	0.9222	0.0000	0.8946	0.9618	0.9639	0.9591	0.9690	0.9668	0.9455	0.9593	0.9459	0.9460	0.9556	0.9583	0.9693	0.9558	0.9592
Misc Politics	0.9295	0.8715	0.9322	0.8801	0.8946	0.0000	0.9547	0.9517	0.9468	0.9601	0.9598	0.9319	0.9472	0.9343	0.9330	0.9427	0.9470	0.9584	0.9441	0.9505
Graphics	0.9569	0.9558	0.9573	0.9520	0.9618	0.9547	0.0000	0.9319	0.9314	0.9405	0.9284	0.9418	0.9393	0.9486	0.9407	0.9508	0.9532	0.9448	0.9544	0.9586
IBM Hardware	0.9543	0.9524	0.9549	0.9489	0.9639	0.9517	0.9319	0.0000	0.9099	0.9336	0.9392	0.9461	0.9260	0.9486	0.9482	0.9405	0.9457	0.9321	0.9474	0.9549
Mac Hardware	0.9522	0.9527	0.9548	0.9476	0.9591	0.9468	0.9314	0.9099	0.0000	0.9431	0.9404	0.9447	0.9263	0.9451	0.9435	0.9391	0.9440	0.9316	0.9447	0.9550
Windows Misc	0.9628	0.9604	0.9642	0.9608	0.9690	0.9601	0.9405	0.9336	0.9431	0.0000	0.9498	0.9552	0.9500	0.9572	0.9578	0.9565	0.9603	0.9536	0.9603	0.9658
Windows X	0.9608	0.9611	0.9618	0.9584	0.9668	0.9598	0.9284	0.9392	0.9404	0.9498	0.0000	0.9482	0.9465	0.9545	0.9523	0.9552	0.9575	0.9544	0.9588	0.9641
Cryptography	0.9417	0.9410	0.9465	0.9267	0.9455	0.9319	0.9418	0.9461	0.9447	0.9552	0.9482	0.0000	0.9409	0.9431	0.9379	0.9470	0.9511	0.9565	0.9513	0.9595
Electronics	0.9501	0.9495	0.9518	0.9462	0.9593	0.9472	0.9393	0.9260	0.9263	0.9500	0.9465	0.9409	0.0000	0.9435	0.9406	0.9335	0.9404	0.9380	0.9463	0.9558
Medicine	0.9475	0.9401	0.9451	0.9336	0.9459	0.9343	0.9486	0.9486	0.9451	0.9572	0.9545	0.9431	0.9435	0.0000	0.9347	0.9452	0.9488	0.9559	0.9481	0.9562
Space	0.9470	0.9364	0.9499	0.9345	0.9460	0.9330	0.9407	0.9482	0.9435	0.9578	0.9523	0.9379	0.9406	0.9347	0.0000	0.9452	0.9511	0.9535	0.9504	0.9566
Autos	0.9494	0.9492	0.9522	0.9410	0.9556	0.9427	0.9508	0.9405	0.9391	0.9565	0.9552	0.9470	0.9335	0.9452	0.9452	0.0000	0.9262	0.9432	0.9424	0.9515
Motorcycles	0.9532	0.9505	0.9546	0.9442	0.9583	0.9470	0.9532	0.9457	0.9440	0.9603	0.9575	0.9511	0.9404	0.9488	0.9511	0.9262	0.0000	0.9531	0.9448	0.9559
For Sale	0.9654	0.9639	0.9644	0.9592	0.9693	0.9584	0.9448	0.9321	0.9316	0.9536	0.9544	0.9565	0.9380	0.9559	0.9535	0.9432	0.9531	0.0000	0.9557	0.9570
Baseball	0.9483	0.9480	0.9505	0.9435	0.9558	0.9441	0.9544	0.9474	0.9447	0.9603	0.9588	0.9513	0.9463	0.9481	0.9504	0.9424	0.9448	0.9557	0.0000	0.9322
Hockey	0.9567	0.9591	0.9594	0.9522	0.9592	0.9505	0.9586	0.9549	0.9550	0.9658	0.9641	0.9595	0.9558	0.9562	0.9566	0.9515	0.9559	0.9570	0.9322	0.0000

Table A.10: WGU Distance between Node Edge Samples (30%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0	17137	19342	19785	22420	21444	20504	18275	17629	24707	22455	20193	18387	21075	21032	18556	18003	18817	18227	20123
Religion	17137	0	20191	19586	23075	19945	21493	19232	18676	25632	23518	21172	19386	21766	21581	19571	18932	19812	19234	21262
Christian	19342	20191	0	20581	23030	22136	21126	18903	18325	25395	23117	20985	19051	21567	21750	19262	18657	19397	18909	20845
Guns	19785	19586	20581	0	22075	19807	20765	18534	17924	25076	22816	20032	18698	20940	20955	18697	18134	19052	18506	20402
Mideast	22420	23075	23030	22075	0	22566	23660	21595	20811	28015	25713	23207	21669	23869	23848	21706	21121	21963	21423	23163
Misc Politics	21444	19945	22136	19807	22566	0	22782	20533	19769	26969	24803	22061	20611	22797	22714	20620	20111	20941	20391	22221
Graphics	20504	21493	21126	20765	23660	22782	0	17685	17123	23845	21255	20417	18213	21345	20986	18832	18227	18277	18675	20427
IBM Hardware	18275	19232	18903	18534	21595	20533	17685	0	14458	21484	19664	18508	15726	19248	19207	16385	15870	15786	16322	18152
Mac Hardware	17629	18676	18325	17924	20811	19769	17123	14458	0	21336	19156	17894	15196	18552	18469	15785	15256	15224	15676	17584
Windows Misc	24707	25632	25395	25076	28015	26969	23845	21484	21336	0	26046	24874	22466	25630	25639	22959	22428	22488	22838	24708
Windows X	22455	23518	23117	22816	25713	24803	21255	19664	19156	26046	0	22430	20220	23366	23241	20771	20166	20402	20628	22458
Cryptography	20193	21172	20985	20032	23207	22061	20417	18508	17894	24874	22430	0	18578	21418	21171	18999	18464	19024	18872	20780
Electronics	18387	19386	19051	18698	21669	20611	18213	15726	15196	22466	20220	18578	0	19314	19177	16411	15964	16234	16552	18452
Medicine	21075	21766	21567	20940	23869	22797	21345	19248	18552	25630	23366	21418	19314	0	21665	19575	19026	19660	19400	21302
Space	21032	21581	21750	20955	23848	22714	20986	19207	18469	25639	23241	21171	19177	21665	0	19548	19087	19541	19463	21295
Autos	18556	19571	19262	18697	21706	20620	18832	16385	15785	22959	20771	18999	16411	19575	19548	0	15701	16599	16615	18491
Motorcycles	18003	18932	18657	18134	21121	20111	18227	15870	15256	22428	20166	18464	15964	19026	19087	15701	0	16234	16012	17954
For Sale	18817	19812	19397	19052	21963	20941	18277	15786	15224	22488	20402	19024	16234	19660	19541	16599	16234	0	16726	18356
Baseball	18227	19234	18909	18506	21423	20391	18675	16322	15676	22838	20628	18872	16552	19400	19463	16615	16012	16726	0	17508
Hockey	20123	21262	20845	20402	23163	22221	20427	18152	17584	24708	22458	20780	18452	21302	21295	18491	17954	18356	17508	0

Table A.11: UGU Distance between Node Edge Samples (30%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.5760	0.8261	0.7935	0.8062	0.7784	0.8122	0.8051	0.8132	0.8065	0.8150	0.8003	0.8101	0.8038	0.8086	0.8067	0.8139	0.8276	0.8105	0.8171
Religion	0.5760	0.0000	0.8194	0.7572	0.8011	0.6665	0.8121	0.8107	0.8127	0.8066	0.8152	0.8051	0.8086	0.7968	0.8077	0.8059	0.8148	0.8319	0.8151	0.8239
Christian	0.8261	0.8194	0.0000	0.8402	0.8482	0.8373	0.8483	0.8543	0.8547	0.8439	0.8434	0.8515	0.8421	0.8320	0.8436	0.8483	0.8585	0.8630	0.8543	0.8622
Guns	0.7935	0.7572	0.8402	0.0000	0.7798	0.6855	0.7982	0.8014	0.7996	0.7903	0.8066	0.7715	0.7947	0.7786	0.7837	0.7887	0.7927	0.8074	0.7940	0.8041
Mideast	0.8062	0.8011	0.8482	0.7798	0.0000	0.7569	0.8693	0.8798	0.8745	0.8726	0.8740	0.8701	0.8762	0.8636	0.8619	0.8735	0.8799	0.8828	0.8729	0.8777
Misc Politics	0.7784	0.6665	0.8373	0.6855	0.7569	0.0000	0.8272	0.8275	0.8273	0.8293	0.8357	0.8119	0.8254	0.8056	0.8136	0.8210	0.8323	0.8419	0.8277	0.8282
Graphics	0.8122	0.8121	0.8483	0.7982	0.8693	0.8272	0.0000	0.7906	0.7987	0.7666	0.7666	0.8130	0.8018	0.8124	0.8013	0.8132	0.8247	0.8073	0.8253	0.8259
IBM Hardware	0.8051	0.8107	0.8543	0.8014	0.8798	0.8275	0.7906	0.0000	0.7094	0.6983	0.7366	0.7648	0.7215	0.7579	0.7707	0.7552	0.7695	0.7287	0.7670	0.7671
Mac Hardware	0.8132	0.8127	0.8547	0.7996	0.8745	0.8273	0.7987	0.7094	0.0000	0.7175	0.7346	0.7524	0.7106	0.7369	0.7470	0.7433	0.7536	0.7234	0.7509	0.7561
Windows Misc	0.8065	0.8066	0.8439	0.7903	0.8726	0.8293	0.7666	0.6983	0.7175	0.0000	0.7470	0.7956	0.7737	0.7823	0.7932	0.7867	0.7945	0.7847	0.7969	0.8020
Windows X	0.8150	0.8152	0.8434	0.8066	0.8740	0.8357	0.7666	0.7366	0.7346	0.7470	0.0000	0.8221	0.8222	0.8372	0.8273	0.8363	0.8397	0.8335	0.8466	0.8516
Cryptography	0.8003	0.8051	0.8515	0.7715	0.8701	0.8119	0.8130	0.7648	0.7524	0.7956	0.8221	0.0000	0.8226	0.8210	0.8200	0.8298	0.8307	0.8400	0.8385	0.8455
Electronics	0.8101	0.8086	0.8421	0.7947	0.8762	0.8254	0.8018	0.7215	0.7106	0.7737	0.8222	0.8226	0.0000	0.7285	0.7391	0.7367	0.7536	0.7363	0.7598	0.7626
Medicine	0.8038	0.7968	0.8320	0.7786	0.8636	0.8056	0.8124	0.7579	0.7369	0.7823	0.8372	0.8210	0.7285	0.0000	0.7963	0.8082	0.8199	0.8127	0.8156	0.8203
Space	0.8086	0.8077	0.8436	0.7837	0.8619	0.8136	0.8013	0.7707	0.7470	0.7932	0.8273	0.8200	0.7391	0.7963	0.0000	0.8253	0.8367	0.8350	0.8310	0.8407
Autos	0.8067	0.8059	0.8483	0.7887	0.8735	0.8210	0.8132	0.7552	0.7433	0.7867	0.8363	0.8298	0.7367	0.8082	0.8253	0.0000	0.7311	0.7509	0.7478	0.7504
Motorcycles	0.8139	0.8148	0.8585	0.7927	0.8799	0.8323	0.8247	0.7695	0.7536	0.7945	0.8397	0.8307	0.7536	0.8199	0.8367	0.7311	0.0000	0.7782	0.7646	0.7761
For Sale	0.8276	0.8319	0.8630	0.8074	0.8828	0.8419	0.8073	0.7287	0.7234	0.7847	0.8335	0.8400	0.7363	0.8127	0.8350	0.7509	0.7782	0.0000	0.7883	0.7927
Baseball	0.8105	0.8151	0.8543	0.7940	0.8729	0.8277	0.8253	0.7670	0.7509	0.7969	0.8466	0.8385	0.7598	0.8156	0.8310	0.7478	0.7646	0.7883	0.0000	0.7433
Hockey	0.8171	0.8239	0.8622	0.8041	0.8777	0.8282	0.8259	0.7671	0.7561	0.8020	0.8516	0.8455	0.7626	0.8203	0.8407	0.7504	0.7761	0.7927	0.7433	0.0000

Table A.12: SID Distance between Node Edge Samples (30%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.9133	0.9695	0.9722	0.9761	0.9756	0.9805	0.9810	0.9797	0.9792	0.9842	0.9733	0.9764	0.9765	0.9789	0.9757	0.9760	0.9846	0.9779	0.9802
Religion	0.9133	0.0000	0.9698	0.9572	0.9742	0.9413	0.9789	0.9786	0.9755	0.9752	0.9843	0.9741	0.9727	0.9727	0.9756	0.9730	0.9741	0.9819	0.9783	0.9791
Christian	0.9695	0.9698	0.0000	0.9789	0.9795	0.9765	0.9825	0.9866	0.9823	0.9829	0.9854	0.9790	0.9771	0.9788	0.9822	0.9792	0.9792	0.9852	0.9821	0.9834
Guns	0.9722	0.9572	0.9789	0.0000	0.9725	0.9502	0.9782	0.9790	0.9787	0.9774	0.9837	0.9697	0.9744	0.9718	0.9757	0.9713	0.9716	0.9818	0.9771	0.9796
Mideast	0.9761	0.9742	0.9795	0.9725	0.0000	0.9596	0.9830	0.9846	0.9819	0.9822	0.9864	0.9765	0.9793	0.9799	0.9806	0.9784	0.9781	0.9865	0.9803	0.9813
Misc Politics	0.9756	0.9413	0.9765	0.9502	0.9596	0.0000	0.9810	0.9821	0.9787	0.9794	0.9852	0.9748	0.9766	0.9734	0.9770	0.9758	0.9779	0.9846	0.9783	0.9817
Graphics	0.9805	0.9789	0.9825	0.9782	0.9830	0.9810	0.0000	0.9754	0.9739	0.9677	0.9723	0.9763	0.9713	0.9767	0.9777	0.9771	0.9749	0.9780	0.9817	0.9819
IBM Hardware	0.9810	0.9786	0.9866	0.9790	0.9846	0.9821	0.9754	0.0000	0.9678	0.9670	0.9794	0.9780	0.9693	0.9788	0.9810	0.9745	0.9740	0.9764	0.9801	0.9826
Mac Hardware	0.9797	0.9755	0.9823	0.9787	0.9819	0.9787	0.9739	0.9678	0.0000	0.9715	0.9779	0.9770	0.9663	0.9752	0.9771	0.9735	0.9710	0.9767	0.9771	0.9816
Windows Misc	0.9792	0.9752	0.9829	0.9774	0.9822	0.9794	0.9677	0.9670	0.9715	0.0000	0.9753	0.9760	0.9700	0.9756	0.9789	0.9760	0.9737	0.9780	0.9787	0.9804
Windows X	0.9842	0.9843	0.9854	0.9837	0.9864	0.9852	0.9723	0.9794	0.9779	0.9753	0.0000	0.9806	0.9774	0.9817	0.9822	0.9813	0.9816	0.9842	0.9837	0.9866
Cryptography	0.9733	0.9741	0.9790	0.9697	0.9765	0.9748	0.9763	0.9780	0.9770	0.9760	0.9806	0.0000	0.9731	0.9764	0.9753	0.9738	0.9749	0.9818	0.9799	0.9822
Electronics	0.9764	0.9727	0.9771	0.9744	0.9793	0.9766	0.9713	0.9693	0.9663	0.9700	0.9774	0.9731	0.0000	0.9706	0.9726	0.9684	0.9665	0.9722	0.9748	0.9781
Medicine	0.9765	0.9727	0.9788	0.9718	0.9799	0.9734	0.9767	0.9788	0.9752	0.9756	0.9817	0.9764	0.9706	0.0000	0.9751	0.9728	0.9732	0.9795	0.9787	0.9793
Space	0.9789	0.9756	0.9822	0.9757	0.9806	0.9770	0.9777	0.9810	0.9771	0.9789	0.9822	0.9753	0.9726	0.9751	0.0000	0.9759	0.9780	0.9832	0.9800	0.9809
Autos	0.9757	0.9730	0.9792	0.9713	0.9784	0.9758	0.9771	0.9745	0.9735	0.9760	0.9813	0.9738	0.9684	0.9728	0.9759	0.0000	0.9625	0.9767	0.9747	0.9784
Motorcycles	0.9760	0.9741	0.9792	0.9716	0.9781	0.9779	0.9749	0.9740	0.9710	0.9737	0.9816	0.9749	0.9665	0.9732	0.9780	0.9625	0.0000	0.9785	0.9745	0.9765
For Sale	0.9846	0.9819	0.9852	0.9818	0.9865	0.9846	0.9780	0.9764	0.9767	0.9780	0.9842	0.9818	0.9722	0.9795	0.9832	0.9767	0.9785	0.0000	0.9817	0.9818
Baseball	0.9779	0.9783	0.9821	0.9771	0.9803	0.9783	0.9817	0.9801	0.9771	0.9787	0.9837	0.9799	0.9748	0.9787	0.9800	0.9747	0.9745	0.9817	0.0000	0.9735
Hockey	0.9802	0.9791	0.9834	0.9796	0.9813	0.9817	0.9819	0.9826	0.9816	0.9804	0.9866	0.9822	0.9781	0.9793	0.9809	0.9784	0.9765	0.9818	0.9735	0.0000

Table A.13: MCS Distance between Random Walk Samples

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.9206	0.9721	0.9741	0.9778	0.9776	0.9815	0.9820	0.9807	0.9802	0.9850	0.9752	0.9776	0.9781	0.9803	0.9771	0.9772	0.9853	0.9791	0.9812
Religion	0.9206	0.0000	0.9724	0.9603	0.9760	0.9459	0.9799	0.9796	0.9767	0.9764	0.9851	0.9759	0.9740	0.9744	0.9773	0.9744	0.9754	0.9828	0.9794	0.9802
Christian	0.9721	0.9724	0.0000	0.9806	0.9812	0.9786	0.9835	0.9874	0.9834	0.9839	0.9862	0.9806	0.9784	0.9803	0.9835	0.9805	0.9804	0.9859	0.9831	0.9844
Guns	0.9741	0.9603	0.9806	0.0000	0.9746	0.9545	0.9794	0.9801	0.9799	0.9786	0.9846	0.9720	0.9759	0.9738	0.9775	0.9731	0.9732	0.9827	0.9785	0.9808
Mideast	0.9778	0.9760	0.9812	0.9746	0.0000	0.9633	0.9839	0.9853	0.9828	0.9831	0.9871	0.9782	0.9803	0.9813	0.9820	0.9796	0.9792	0.9872	0.9813	0.9823
Misc Politics	0.9776	0.9459	0.9786	0.9545	0.9633	0.0000	0.9821	0.9831	0.9799	0.9806	0.9860	0.9769	0.9780	0.9755	0.9789	0.9773	0.9792	0.9854	0.9796	0.9828
Graphics	0.9815	0.9799	0.9835	0.9794	0.9839	0.9821	0.0000	0.9769	0.9756	0.9697	0.9741	0.9779	0.9730	0.9781	0.9792	0.9784	0.9761	0.9792	0.9826	0.9828
IBM Hardware	0.9820	0.9796	0.9874	0.9801	0.9853	0.9831	0.9769	0.0000	0.9702	0.9689	0.9806	0.9794	0.9713	0.9800	0.9822	0.9760	0.9753	0.9779	0.9811	0.9835
Mac Hardware	0.9807	0.9767	0.9834	0.9799	0.9828	0.9799	0.9756	0.9702	0.0000	0.9732	0.9792	0.9785	0.9685	0.9766	0.9786	0.9751	0.9725	0.9782	0.9782	0.9825
Windows Misc	0.9802	0.9764	0.9839	0.9786	0.9831	0.9806	0.9697	0.9689	0.9732	0.0000	0.9768	0.9774	0.9716	0.9770	0.9801	0.9773	0.9748	0.9792	0.9797	0.9813
Windows X	0.9850	0.9851	0.9862	0.9846	0.9871	0.9860	0.9741	0.9806	0.9792	0.9768	0.0000	0.9819	0.9786	0.9827	0.9833	0.9823	0.9824	0.9850	0.9844	0.9872
Cryptography	0.9752	0.9759	0.9806	0.9720	0.9782	0.9769	0.9779	0.9794	0.9785	0.9774	0.9819	0.0000	0.9748	0.9782	0.9773	0.9754	0.9763	0.9828	0.9810	0.9832
Electronics	0.9776	0.9740	0.9784	0.9759	0.9803	0.9780	0.9730	0.9713	0.9685	0.9716	0.9786	0.9748	0.0000	0.9723	0.9743	0.9703	0.9682	0.9739	0.9761	0.9792
Medicine	0.9781	0.9744	0.9803	0.9738	0.9813	0.9755	0.9781	0.9800	0.9766	0.9770	0.9827	0.9782	0.9723	0.0000	0.9770	0.9745	0.9746	0.9807	0.9798	0.9805
Space	0.9803	0.9773	0.9835	0.9775	0.9820	0.9789	0.9792	0.9822	0.9786	0.9801	0.9833	0.9773	0.9743	0.9770	0.0000	0.9775	0.9793	0.9842	0.9811	0.9820
Autos	0.9771	0.9744	0.9805	0.9731	0.9796	0.9773	0.9784	0.9760	0.9751	0.9773	0.9823	0.9754	0.9703	0.9745	0.9775	0.0000	0.9649	0.9781	0.9761	0.9796
Motorcycles	0.9772	0.9754	0.9804	0.9732	0.9792	0.9792	0.9761	0.9753	0.9725	0.9748	0.9824	0.9763	0.9682	0.9746	0.9793	0.9649	0.0000	0.9796	0.9758	0.9776
For Sale	0.9853	0.9828	0.9859	0.9827	0.9872	0.9854	0.9792	0.9779	0.9782	0.9792	0.9850	0.9828	0.9739	0.9807	0.9842	0.9781	0.9796	0.0000	0.9826	0.9828
Baseball	0.9791	0.9794	0.9831	0.9785	0.9813	0.9796	0.9826	0.9811	0.9782	0.9797	0.9844	0.9810	0.9761	0.9798	0.9811	0.9761	0.9758	0.9826	0.0000	0.9752
Hockey	0.9812	0.9802	0.9844	0.9808	0.9823	0.9828	0.9828	0.9835	0.9825	0.9813	0.9872	0.9832	0.9792	0.9805	0.9820	0.9796	0.9776	0.9828	0.9752	0.0000

Table A.14: WGU Distance between Random Walk Samples

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0	19815	21532	21471	24194	23612	21540	19315	18665	25582	23569	21595	19429	22407	22482	19615	18891	19581	19387	21135
Religion	19815	0	22613	21930	25185	23177	22557	20310	19594	26467	24672	22704	20364	23318	23425	20586	19900	20574	20484	22180
Christian	21532	22613	0	22387	24996	24294	22266	20171	19405	26412	24273	22467	20093	23145	23266	20385	19647	20249	20185	21911
Guns	21471	21930	22387	0	24523	23011	21935	19732	19120	25987	24044	21940	19844	22696	22839	19940	19220	19970	19848	21604
Mideast	24194	25185	24996	24523	0	25910	24730	22541	21821	28812	26775	24777	22599	25621	25634	22775	22023	22761	22545	24261
Misc Politics	23612	23177	24294	23011	25910	0	24066	21865	21123	28094	26137	24145	21923	24757	24900	22101	21453	22103	21897	23709
Graphics	21540	22557	22266	21935	24730	24066	0	19357	18709	25280	23297	21947	19485	22645	22666	19901	19083	19581	19761	21441
IBM Hardware	19315	20310	20171	19732	22541	21865	19357	0	16318	23059	21364	19784	17222	20494	20559	17592	16840	17304	17462	19220
Mac Hardware	18665	19594	19405	19120	21821	21123	18709	16318	0	22661	20704	19148	16538	19758	19811	16964	16152	16714	16764	18580
Windows Misc	25582	26467	26412	25987	28812	28094	25280	23059	22661	0	27493	26005	23461	26665	26808	23933	23091	23671	23743	25487
Windows X	23569	24672	24273	24044	26775	26137	23297	21364	20704	27493	0	23996	21562	24726	24729	21928	21198	21690	21714	23522
Cryptography	21595	22704	22467	21940	24777	24145	21947	19784	19148	26005	23996	0	19884	22976	22909	20112	19420	20056	20032	21792
Electronics	19429	20364	20093	19844	22599	21923	19485	17222	16538	23461	21562	19884	0	20460	20517	17670	16878	17446	17568	19340
Medicine	22407	23318	23145	22696	25621	24757	22645	20494	19758	26665	24726	22976	20460	0	23581	20754	20034	20658	20670	22362
Space	22482	23425	23266	22839	25634	24900	22666	20559	19811	26808	24729	22909	20517	23581	0	20855	20197	20777	20699	22405
Autos	19615	20586	20385	19940	22775	22101	19901	17592	16964	23933	21928	20112	17670	20754	20855	0	16966	17800	17774	19562
Motorcycles	18891	19900	19647	19220	22023	21453	19083	16840	16152	23091	21198	19420	16878	20034	20197	16966	0	17120	17036	18754
For Sale	19581	20574	20249	19970	22761	22103	19581	17304	16714	23671	21690	20056	17446	20658	20777	17800	17120	0	17652	19328
Baseball	19387	20484	20185	19848	22545	21897	19761	17462	16764	23743	21714	20032	17568	20670	20699	17774	17036	17652	0	19086
Hockey	21135	22180	21911	21604	24261	23709	21441	19220	18580	25487	23522	21792	19340	22362	22405	19562	18754	19328	19086	0

Table A.15: UGU Distance between Random Walk Samples

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.6991	0.8586	0.8258	0.8253	0.8302	0.8328	0.8355	0.8415	0.8124	0.8504	0.8212	0.8234	0.8176	0.8167	0.8123	0.8192	0.8584	0.8429	0.8545
Religion	0.6991	0.0000	0.8301	0.7749	0.7898	0.7376	0.7994	0.8097	0.7992	0.7666	0.8243	0.7995	0.7723	0.7642	0.8034	0.7790	0.7757	0.8138	0.7912	0.8010
Christian	0.8586	0.8301	0.0000	0.8284	0.8060	0.8248	0.8217	0.8444	0.8160	0.7990	0.8535	0.8456	0.8104	0.8028	0.8249	0.8220	0.8155	0.8431	0.8348	0.8341
Guns	0.8258	0.7749	0.8284	0.0000	0.7856	0.7327	0.8088	0.7849	0.8059	0.7585	0.8343	0.7715	0.7807	0.7710	0.7858	0.7807	0.7761	0.8184	0.7899	0.8206
Mideast	0.8253	0.7898	0.8060	0.7856	0.0000	0.8016	0.8683	0.8809	0.8715	0.8533	0.8936	0.8649	0.8560	0.8467	0.8720	0.8560	0.8528	0.8714	0.8621	0.8809
Misc Politics	0.8302	0.7376	0.8248	0.7327	0.8016	0.0000	0.8458	0.8355	0.8452	0.8062	0.8832	0.8230	0.8193	0.7965	0.8302	0.8214	0.8204	0.8379	0.8372	0.8459
Graphics	0.8328	0.7994	0.8217	0.8088	0.8683	0.8458	0.0000	0.8269	0.8319	0.7755	0.8192	0.8395	0.8072	0.8116	0.8257	0.8240	0.8195	0.8127	0.8473	0.8471
IBM Hardware	0.8355	0.8097	0.8444	0.7849	0.8809	0.8355	0.8269	0.0000	0.7461	0.6801	0.8200	0.7535	0.7185	0.7375	0.7485	0.7260	0.7266	0.7378	0.7337	0.7587
Mac Hardware	0.8415	0.7992	0.8160	0.8059	0.8715	0.8452	0.8319	0.7461	0.0000	0.7207	0.8029	0.7685	0.7519	0.7535	0.7638	0.7504	0.7527	0.7689	0.7731	0.7737
Windows Misc	0.8124	0.7666	0.7990	0.7585	0.8533	0.8062	0.7755	0.6801	0.7207	0.0000	0.8216	0.8028	0.7883	0.7899	0.8024	0.7938	0.7972	0.8107	0.8106	0.8184
Windows X	0.8504	0.8243	0.8535	0.8343	0.8936	0.8832	0.8192	0.8200	0.8029	0.8216	0.0000	0.8542	0.8391	0.8616	0.8548	0.8695	0.8579	0.8558	0.8645	0.8893
Cryptography	0.8212	0.7995	0.8456	0.7715	0.8649	0.8230	0.8395	0.7535	0.7685	0.8028	0.8542	0.0000	0.8416	0.8401	0.8428	0.8451	0.8510	0.8764	0.8585	0.8806
Electronics	0.8234	0.7723	0.8104	0.7807	0.8560	0.8193	0.8072	0.7185	0.7519	0.7883	0.8391	0.8416	0.0000	0.7163	0.7577	0.7345	0.7332	0.7517	0.7508	0.7771
Medicine	0.8176	0.7642	0.8028	0.7710	0.8467	0.7965	0.8116	0.7375	0.7535	0.7899	0.8616	0.8401	0.7163	0.0000	0.7895	0.7886	0.7903	0.8008	0.8113	0.8035
Space	0.8167	0.8034	0.8249	0.7858	0.8720	0.8302	0.8257	0.7485	0.7638	0.8024	0.8548	0.8428	0.7577	0.7895	0.0000	0.8094	0.8202	0.8321	0.8253	0.8494
Autos	0.8123	0.7790	0.8220	0.7807	0.8560	0.8214	0.8240	0.7260	0.7504	0.7938	0.8695	0.8451	0.7345	0.7886	0.8094	0.0000	0.7471	0.7752	0.7906	0.7988
Motorcycles	0.8192	0.7757	0.8155	0.7761	0.8528	0.8204	0.8195	0.7266	0.7527	0.7972	0.8579	0.8510	0.7332	0.7903	0.8202	0.7471	0.0000	0.7401	0.7359	0.7362
For Sale	0.8584	0.8138	0.8431	0.8184	0.8714	0.8379	0.8127	0.7378	0.7689	0.8107	0.8558	0.8764	0.7517	0.8008	0.8321	0.7752	0.7401	0.0000	0.8202	0.8070
Baseball	0.8429	0.7912	0.8348	0.7899	0.8621	0.8372	0.8473	0.7337	0.7731	0.8106	0.8645	0.8585	0.7508	0.8113	0.8253	0.7906	0.7359	0.8202	0.0000	0.7785
Hockey	0.8545	0.8010	0.8341	0.8206	0.8809	0.8459	0.8471	0.7587	0.7737	0.8184	0.8893	0.8806	0.7771	0.8035	0.8494	0.7988	0.7362	0.8070	0.7785	0.0000

Table A.16: SID Distance between Random Walk Samples

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.8315	0.9288	0.9397	0.9476	0.9357	0.9622	0.9607	0.9602	0.9700	0.9672	0.9481	0.9563	0.9517	0.9530	0.9545	0.9574	0.9692	0.9559	0.9626
Religion	0.8315	0.0000	0.9247	0.9058	0.9391	0.8600	0.9615	0.9596	0.9582	0.9687	0.9668	0.9471	0.9567	0.9433	0.9401	0.9533	0.9559	0.9677	0.9558	0.9642
Christian	0.9288	0.9247	0.0000	0.9458	0.9491	0.9401	0.9619	0.9611	0.9604	0.9711	0.9670	0.9525	0.9580	0.9509	0.9562	0.9565	0.9601	0.9688	0.9568	0.9641
Guns	0.9397	0.9058	0.9458	0.0000	0.9258	0.8744	0.9569	0.9549	0.9540	0.9670	0.9641	0.9310	0.9519	0.9367	0.9396	0.9444	0.9493	0.9629	0.9492	0.9566
Mideast	0.9476	0.9391	0.9491	0.9258	0.0000	0.8916	0.9660	0.9683	0.9651	0.9748	0.9725	0.9525	0.9656	0.9502	0.9515	0.9608	0.9642	0.9727	0.9617	0.9643
Misc Politics	0.9357	0.8600	0.9401	0.8744	0.8916	0.0000	0.9596	0.9586	0.9546	0.9672	0.9654	0.9381	0.9536	0.9377	0.9390	0.9489	0.9539	0.9636	0.9508	0.9559
Graphics	0.9622	0.9615	0.9619	0.9569	0.9660	0.9596	0.0000	0.9355	0.9374	0.9478	0.9354	0.9481	0.9465	0.9514	0.9484	0.9541	0.9580	0.9498	0.9594	0.9624
IBM Hardware	0.9607	0.9596	0.9611	0.9549	0.9683	0.9586	0.9355	0.0000	0.9170	0.9410	0.9463	0.9515	0.9324	0.9522	0.9536	0.9453	0.9531	0.9377	0.9537	0.9594
Mac Hardware	0.9602	0.9582	0.9604	0.9540	0.9651	0.9546	0.9374	0.9170	0.0000	0.9519	0.9483	0.9512	0.9325	0.9524	0.9513	0.9459	0.9502	0.9374	0.9510	0.9603
Windows Misc	0.9700	0.9687	0.9711	0.9670	0.9748	0.9672	0.9478	0.9410	0.9519	0.0000	0.9563	0.9626	0.9586	0.9653	0.9652	0.9626	0.9666	0.9599	0.9674	0.9721
Windows X	0.9672	0.9668	0.9670	0.9641	0.9725	0.9654	0.9354	0.9463	0.9483	0.9563	0.0000	0.9545	0.9534	0.9600	0.9593	0.9611	0.9630	0.9595	0.9647	0.9691
Cryptography	0.9481	0.9471	0.9525	0.9310	0.9525	0.9381	0.9481	0.9515	0.9512	0.9626	0.9545	0.0000	0.9470	0.9486	0.9456	0.9522	0.9561	0.9615	0.9554	0.9638
Electronics	0.9563	0.9567	0.9580	0.9519	0.9656	0.9536	0.9465	0.9324	0.9325	0.9586	0.9534	0.9470	0.0000	0.9502	0.9467	0.9394	0.9458	0.9414	0.9529	0.9614
Medicine	0.9517	0.9433	0.9509	0.9367	0.9502	0.9377	0.9514	0.9522	0.9524	0.9653	0.9600	0.9486	0.9502	0.0000	0.9391	0.9493	0.9547	0.9603	0.9549	0.9597
Space	0.9530	0.9401	0.9562	0.9396	0.9515	0.9390	0.9484	0.9536	0.9513	0.9652	0.9593	0.9456	0.9467	0.9391	0.0000	0.9524	0.9564	0.9600	0.9573	0.9605
Autos	0.9545	0.9533	0.9565	0.9444	0.9608	0.9489	0.9541	0.9453	0.9459	0.9626	0.9611	0.9522	0.9394	0.9493	0.9524	0.0000	0.9318	0.9467	0.9484	0.9579
Motorcycles	0.9574	0.9559	0.9601	0.9493	0.9642	0.9539	0.9580	0.9531	0.9502	0.9666	0.9630	0.9561	0.9458	0.9547	0.9564	0.9318	0.0000	0.9571	0.9516	0.9608
For Sale	0.9692	0.9677	0.9688	0.9629	0.9727	0.9636	0.9498	0.9377	0.9374	0.9599	0.9595	0.9615	0.9414	0.9603	0.9600	0.9467	0.9571	0.0000	0.9606	0.9620
Baseball	0.9559	0.9558	0.9568	0.9492	0.9617	0.9508	0.9594	0.9537	0.9510	0.9674	0.9647	0.9554	0.9529	0.9549	0.9573	0.9484	0.9516	0.9606	0.0000	0.9386
Hockey	0.9626	0.9642	0.9641	0.9566	0.9643	0.9559	0.9624	0.9594	0.9603	0.9721	0.9691	0.9638	0.9614	0.9597	0.9605	0.9579	0.9608	0.9620	0.9386	0.0000

Table A.17: MCS Distance between Node-Edge Samples (50%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.8392	0.9345	0.9441	0.9514	0.9405	0.9642	0.9627	0.9624	0.9712	0.9688	0.9517	0.9588	0.9551	0.9562	0.9573	0.9600	0.9707	0.9585	0.9645
Religion	0.8392	0.0000	0.9309	0.9115	0.9435	0.8662	0.9636	0.9617	0.9603	0.9699	0.9684	0.9507	0.9591	0.9469	0.9437	0.9562	0.9587	0.9693	0.9583	0.9662
Christian	0.9345	0.9309	0.0000	0.9499	0.9531	0.9448	0.9640	0.9631	0.9625	0.9723	0.9687	0.9559	0.9604	0.9544	0.9593	0.9592	0.9627	0.9703	0.9594	0.9661
Guns	0.9441	0.9115	0.9499	0.0000	0.9310	0.8815	0.9594	0.9573	0.9565	0.9684	0.9658	0.9356	0.9548	0.9409	0.9436	0.9480	0.9526	0.9649	0.9523	0.9591
Mideast	0.9514	0.9435	0.9531	0.9310	0.0000	0.8981	0.9678	0.9699	0.9668	0.9758	0.9738	0.9558	0.9675	0.9534	0.9548	0.9632	0.9664	0.9740	0.9640	0.9662
Misc Politics	0.9405	0.8662	0.9448	0.8815	0.8981	0.0000	0.9620	0.9607	0.9569	0.9685	0.9672	0.9425	0.9564	0.9419	0.9432	0.9521	0.9568	0.9655	0.9537	0.9584
Graphics	0.9642	0.9636	0.9640	0.9594	0.9678	0.9620	0.0000	0.9389	0.9409	0.9498	0.9387	0.9513	0.9496	0.9543	0.9516	0.9566	0.9603	0.9524	0.9615	0.9642
IBM Hardware	0.9627	0.9617	0.9631	0.9573	0.9699	0.9607	0.9389	0.0000	0.9215	0.9433	0.9488	0.9542	0.9362	0.9548	0.9563	0.9483	0.9557	0.9408	0.9559	0.9613
Mac Hardware	0.9624	0.9603	0.9625	0.9565	0.9668	0.9569	0.9409	0.9215	0.0000	0.9538	0.9508	0.9540	0.9364	0.9552	0.9540	0.9489	0.9530	0.9405	0.9534	0.9622
Windows Misc	0.9712	0.9699	0.9723	0.9684	0.9758	0.9685	0.9498	0.9433	0.9538	0.0000	0.9579	0.9642	0.9603	0.9667	0.9667	0.9641	0.9679	0.9614	0.9685	0.9731
Windows X	0.9688	0.9684	0.9687	0.9658	0.9738	0.9672	0.9387	0.9488	0.9508	0.9579	0.0000	0.9569	0.9557	0.9620	0.9615	0.9629	0.9647	0.9612	0.9662	0.9705
Cryptography	0.9517	0.9507	0.9559	0.9356	0.9558	0.9425	0.9513	0.9542	0.9540	0.9642	0.9569	0.0000	0.9503	0.9521	0.9493	0.9552	0.9588	0.9635	0.9579	0.9657
Electronics	0.9588	0.9591	0.9604	0.9548	0.9675	0.9564	0.9496	0.9362	0.9364	0.9603	0.9557	0.9503	0.0000	0.9533	0.9501	0.9430	0.9491	0.9447	0.9553	0.9633
Medicine	0.9551	0.9469	0.9544	0.9409	0.9534	0.9419	0.9543	0.9548	0.9552	0.9667	0.9620	0.9521	0.9533	0.0000	0.9431	0.9525	0.9576	0.9625	0.9575	0.9619
Space	0.9562	0.9437	0.9593	0.9436	0.9548	0.9432	0.9516	0.9563	0.9540	0.9667	0.9615	0.9493	0.9501	0.9431	0.0000	0.9556	0.9593	0.9623	0.9598	0.9628
Autos	0.9573	0.9562	0.9592	0.9480	0.9632	0.9521	0.9566	0.9483	0.9489	0.9641	0.9629	0.9552	0.9430	0.9525	0.9556	0.0000	0.9363	0.9495	0.9512	0.9601
Motorcycles	0.9600	0.9587	0.9627	0.9526	0.9664	0.9568	0.9603	0.9557	0.9530	0.9679	0.9647	0.9588	0.9491	0.9576	0.9593	0.9363	0.0000	0.9594	0.9544	0.9631
For Sale	0.9707	0.9693	0.9703	0.9649	0.9740	0.9655	0.9524	0.9408	0.9405	0.9614	0.9612	0.9635	0.9447	0.9625	0.9623	0.9495	0.9594	0.0000	0.9626	0.9639
Baseball	0.9585	0.9583	0.9594	0.9523	0.9640	0.9537	0.9615	0.9559	0.9534	0.9685	0.9662	0.9579	0.9553	0.9575	0.9598	0.9512	0.9544	0.9626	0.0000	0.9422
Hockey	0.9645	0.9662	0.9661	0.9591	0.9662	0.9584	0.9642	0.9613	0.9622	0.9731	0.9705	0.9657	0.9633	0.9619	0.9628	0.9601	0.9631	0.9639	0.9422	0.0000

Table A.18: WGU Distance between Node-Edge Samples (50%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0	27995	33280	33696	38251	36534	34678	30976	29988	41878	38035	34341	31185	35667	35705	31423	30418	31695	31006	34068
Religion	27995	0	34681	33139	39326	32885	36387	32657	31607	43545	39776	35978	32942	36778	36504	33082	32071	33380	32731	35943
Christian	33280	34681	0	35092	39385	37842	35692	32028	31022	43014	39059	35641	32305	36623	36939	32559	31606	32713	32078	35210
Guns	33696	33139	35092	0	37449	33108	35120	31418	30410	42438	38599	33989	31711	35419	35569	31607	30740	32121	31390	34480
Mideast	38251	39326	39385	37449	0	37879	39909	36429	35229	47329	43458	39486	36714	40386	40458	36730	35777	36954	36293	39153
Misc Politics	36534	32885	37842	33108	37879	0	38524	34846	33626	45708	41955	37559	34991	38579	38637	35019	34188	35401	34648	37626
Graphics	34678	36387	35692	35120	39909	38524	0	29896	29090	40494	36167	34687	30987	35981	35745	31753	30816	30929	31572	34430
IBM Hardware	30976	32657	32028	31418	36429	34846	29896	0	24668	36506	33415	31353	26755	32477	32533	27735	26986	26775	27674	30642
Mac Hardware	29988	31607	31022	30410	35229	33626	29090	24668	0	36332	32601	30387	25847	31549	31435	26829	25890	25833	26586	29732
Windows Misc	41878	43545	43014	42438	47329	45708	40494	36506	36332	0	44129	42209	38223	43545	43499	38851	37958	38071	38698	41784
Windows X	38035	39776	39059	38599	43458	41955	36167	33415	32601	44129	0	38042	34324	39540	39456	35160	34101	34472	34893	37911
Cryptography	34341	35978	35641	33989	39486	37559	34687	31353	30387	42209	38042	0	31550	36346	36100	32192	31249	32156	31873	35069
Electronics	31185	32942	32305	31711	36714	34991	30987	26755	25847	38223	34324	31550	0	32826	32578	27880	27073	27422	28087	31225
Medicine	35667	36778	36623	35419	40386	38579	35981	32477	31549	43545	39540	36346	32826	0	36718	33104	32273	33202	32947	35907
Space	35705	36504	36939	35569	40458	38637	35745	32533	31435	43499	39456	36100	32578	36718	0	33266	32339	33146	33053	35929
Autos	31423	33082	32559	31607	36730	35019	31753	27735	26829	38851	35160	32192	27880	33104	33266	0	26703	28014	28187	31357
Motorcycles	30418	32071	31606	30740	35777	34188	30816	26986	25890	37958	34101	31249	27073	32273	32339	26703	0	27403	27206	30360
For Sale	31695	33380	32713	32121	36954	35401	30929	26775	25833	38071	34472	32156	27422	33202	33146	28014	27403	0	28265	31019
Baseball	31006	32731	32078	31390	36293	34648	31572	27674	26586	38698	34893	31873	28087	32947	33053	28187	27206	28265	0	29772
Hockey	34068	35943	35210	34480	39153	37626	34430	30642	29732	41784	37911	35069	31225	35907	35929	31357	30360	31019	29772	0

Table A.19: UGU Distance between Node-Edge Samples (50%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.5967	0.8478	0.8234	0.8393	0.8095	0.8388	0.8380	0.8415	0.8395	0.8479	0.8304	0.8366	0.8329	0.8336	0.8356	0.8442	0.8546	0.8382	0.8459
Religion	0.5967	0.0000	0.8426	0.7829	0.8310	0.6782	0.8428	0.8422	0.8408	0.8391	0.8484	0.8360	0.8383	0.8298	0.8325	0.8388	0.8438	0.8550	0.8449	0.8484
Christian	0.8478	0.8426	0.0000	0.8658	0.8698	0.8647	0.8737	0.8748	0.8755	0.8702	0.8706	0.8711	0.8660	0.8580	0.8708	0.8713	0.8796	0.8798	0.8785	0.8789
Guns	0.8234	0.7829	0.8658	0.0000	0.8063	0.7024	0.8304	0.8321	0.8322	0.8259	0.8357	0.8076	0.8242	0.8114	0.8148	0.8170	0.8244	0.8375	0.8319	0.8331
Mideast	0.8393	0.8310	0.8698	0.8063	0.0000	0.7705	0.8915	0.8996	0.8930	0.8952	0.8950	0.8905	0.8935	0.8803	0.8803	0.8931	0.8983	0.9011	0.8957	0.8983
Misc Politics	0.8095	0.6782	0.8647	0.7024	0.7705	0.0000	0.8568	0.8567	0.8524	0.8565	0.8632	0.8377	0.8499	0.8349	0.8404	0.8516	0.8575	0.8627	0.8537	0.8555
Graphics	0.8388	0.8428	0.8737	0.8304	0.8915	0.8568	0.0000	0.8292	0.8375	0.8119	0.8082	0.8445	0.8359	0.8405	0.8418	0.8473	0.8582	0.8463	0.8597	0.8592
IBM Hardware	0.8380	0.8422	0.8748	0.8321	0.8996	0.8567	0.8292	0.0000	0.7574	0.7423	0.7842	0.8070	0.7605	0.7960	0.8065	0.7935	0.8100	0.7703	0.8078	0.8083
Mac Hardware	0.8415	0.8408	0.8755	0.8322	0.8930	0.8524	0.8375	0.7574	0.0000	0.7685	0.7773	0.7952	0.7613	0.7865	0.7787	0.7857	0.7937	0.7650	0.7926	0.8003
Windows Misc	0.8395	0.8391	0.8702	0.8259	0.8952	0.8565	0.8119	0.7423	0.7685	0.0000	0.7938	0.8290	0.8197	0.8265	0.8302	0.8240	0.8346	0.8224	0.8362	0.8373
Windows X	0.8479	0.8484	0.8706	0.8357	0.8950	0.8632	0.8082	0.7842	0.7773	0.7938	0.0000	0.8519	0.8528	0.8647	0.8613	0.8644	0.8705	0.8607	0.8747	0.8773
Cryptography	0.8304	0.8360	0.8711	0.8076	0.8905	0.8377	0.8445	0.8070	0.7952	0.8290	0.8519	0.0000	0.8476	0.8463	0.8469	0.8557	0.8573	0.8639	0.8638	0.8705
Electronics	0.8366	0.8383	0.8660	0.8242	0.8935	0.8499	0.8359	0.7605	0.7613	0.8197	0.8528	0.8476	0.0000	0.7790	0.7882	0.7799	0.7935	0.7803	0.8004	0.8059
Medicine	0.8329	0.8298	0.8580	0.8114	0.8803	0.8349	0.8405	0.7960	0.7865	0.8265	0.8647	0.8463	0.7790	0.0000	0.8324	0.8416	0.8528	0.8485	0.8479	0.8559
Space	0.8336	0.8325	0.8708	0.8148	0.8803	0.8404	0.8418	0.8065	0.7787	0.8302	0.8613	0.8469	0.7882	0.8324	0.0000	0.8523	0.8599	0.8629	0.8602	0.8677
Autos	0.8356	0.8388	0.8713	0.8170	0.8931	0.8516	0.8473	0.7935	0.7857	0.8240	0.8644	0.8557	0.7799	0.8416	0.8523	0.0000	0.7789	0.7920	0.7886	0.7992
Motorcycles	0.8442	0.8438	0.8796	0.8244	0.8983	0.8575	0.8582	0.8100	0.7937	0.8346	0.8705	0.8573	0.7935	0.8528	0.8599	0.7789	0.0000	0.8111	0.8051	0.8131
For Sale	0.8546	0.8550	0.8798	0.8375	0.9011	0.8627	0.8463	0.7703	0.7650	0.8224	0.8607	0.8639	0.7803	0.8485	0.8629	0.7920	0.8111	0.0000	0.8249	0.8259
Baseball	0.8382	0.8449	0.8785	0.8319	0.8957	0.8537	0.8597	0.8078	0.7926	0.8362	0.8747	0.8638	0.8004	0.8479	0.8602	0.7886	0.8051	0.8249	0.0000	0.7796
Hockey	0.8459	0.8484	0.8789	0.8331	0.8983	0.8555	0.8592	0.8083	0.8003	0.8373	0.8773	0.8705	0.8059	0.8559	0.8677	0.7992	0.8131	0.8259	0.7796	0.0000

Table A.20: SID Distance between Node-Edge Samples (50%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.8685	0.9624	0.9683	0.9726	0.9671	0.9777	0.9755	0.9754	0.9775	0.9804	0.9712	0.9721	0.9732	0.9743	0.9744	0.9757	0.9813	0.9735	0.9785
Religion	0.8685	0.0000	0.9594	0.9449	0.9702	0.9127	0.9765	0.9734	0.9733	0.9747	0.9802	0.9686	0.9715	0.9675	0.9702	0.9706	0.9738	0.9795	0.9723	0.9788
Christian	0.9624	0.9594	0.0000	0.9731	0.9751	0.9713	0.9792	0.9771	0.9780	0.9797	0.9826	0.9751	0.9739	0.9738	0.9778	0.9756	0.9779	0.9840	0.9758	0.9809
Guns	0.9683	0.9449	0.9731	0.0000	0.9668	0.9326	0.9769	0.9742	0.9731	0.9757	0.9803	0.9641	0.9702	0.9670	0.9699	0.9688	0.9714	0.9790	0.9706	0.9768
Mideast	0.9726	0.9702	0.9751	0.9668	0.0000	0.9470	0.9821	0.9824	0.9809	0.9819	0.9844	0.9762	0.9794	0.9755	0.9764	0.9781	0.9806	0.9857	0.9777	0.9815
Misc Politics	0.9671	0.9127	0.9713	0.9326	0.9470	0.0000	0.9785	0.9757	0.9748	0.9765	0.9818	0.9666	0.9724	0.9672	0.9687	0.9710	0.9746	0.9806	0.9724	0.9785
Graphics	0.9777	0.9765	0.9792	0.9769	0.9821	0.9785	0.0000	0.9637	0.9657	0.9627	0.9638	0.9739	0.9662	0.9728	0.9717	0.9716	0.9752	0.9734	0.9763	0.9792
IBM Hardware	0.9755	0.9734	0.9771	0.9742	0.9824	0.9757	0.9637	0.0000	0.9519	0.9560	0.9721	0.9725	0.9570	0.9722	0.9734	0.9653	0.9714	0.9659	0.9716	0.9757
Mac Hardware	0.9754	0.9733	0.9780	0.9731	0.9809	0.9748	0.9657	0.9519	0.0000	0.9653	0.9713	0.9711	0.9559	0.9708	0.9710	0.9655	0.9701	0.9654	0.9708	0.9757
Windows Misc	0.9775	0.9747	0.9797	0.9757	0.9819	0.9765	0.9627	0.9560	0.9653	0.0000	0.9697	0.9743	0.9668	0.9739	0.9747	0.9713	0.9754	0.9730	0.9754	0.9796
Windows X	0.9804	0.9802	0.9826	0.9803	0.9844	0.9818	0.9638	0.9721	0.9713	0.9697	0.0000	0.9760	0.9711	0.9778	0.9767	0.9764	0.9781	0.9790	0.9802	0.9821
Cryptography	0.9712	0.9686	0.9751	0.9641	0.9762	0.9666	0.9739	0.9725	0.9711	0.9743	0.9760	0.0000	0.9677	0.9716	0.9703	0.9713	0.9744	0.9784	0.9739	0.9795
Electronics	0.9721	0.9715	0.9739	0.9702	0.9794	0.9724	0.9662	0.9570	0.9559	0.9668	0.9711	0.9677	0.0000	0.9677	0.9671	0.9591	0.9660	0.9657	0.9689	0.9747
Medicine	0.9732	0.9675	0.9738	0.9670	0.9755	0.9672	0.9728	0.9722	0.9708	0.9739	0.9778	0.9716	0.9677	0.0000	0.9673	0.9702	0.9726	0.9763	0.9722	0.9781
Space	0.9743	0.9702	0.9778	0.9699	0.9764	0.9687	0.9717	0.9734	0.9710	0.9747	0.9767	0.9703	0.9671	0.9673	0.0000	0.9723	0.9755	0.9789	0.9755	0.9789
Autos	0.9744	0.9706	0.9756	0.9688	0.9781	0.9710	0.9716	0.9653	0.9655	0.9713	0.9764	0.9713	0.9591	0.9702	0.9723	0.0000	0.9615	0.9678	0.9676	0.9742
Motorcycles	0.9757	0.9738	0.9779	0.9714	0.9806	0.9746	0.9752	0.9714	0.9701	0.9754	0.9781	0.9744	0.9660	0.9726	0.9755	0.9615	0.0000	0.9758	0.9708	0.9773
For Sale	0.9813	0.9795	0.9840	0.9790	0.9857	0.9806	0.9734	0.9659	0.9654	0.9730	0.9790	0.9784	0.9657	0.9763	0.9789	0.9678	0.9758	0.0000	0.9758	0.9786
Baseball	0.9735	0.9723	0.9758	0.9706	0.9777	0.9724	0.9763	0.9716	0.9708	0.9754	0.9802	0.9739	0.9689	0.9722	0.9755	0.9676	0.9708	0.9758	0.0000	0.9656
Hockey	0.9785	0.9788	0.9809	0.9768	0.9815	0.9785	0.9792	0.9757	0.9757	0.9796	0.9821	0.9795	0.9747	0.9781	0.9789	0.9742	0.9773	0.9786	0.9656	0.0000

Table A.21: MCS Distance between Random Walk Samples (50%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.8793	0.9661	0.9709	0.9750	0.9699	0.9790	0.9768	0.9767	0.9786	0.9815	0.9735	0.9737	0.9752	0.9762	0.9760	0.9772	0.9822	0.9752	0.9797
Religion	0.8793	0.0000	0.9633	0.9494	0.9727	0.9197	0.9778	0.9748	0.9747	0.9760	0.9813	0.9709	0.9731	0.9698	0.9724	0.9724	0.9754	0.9805	0.9740	0.9799
Christian	0.9661	0.9633	0.0000	0.9755	0.9774	0.9740	0.9805	0.9785	0.9794	0.9808	0.9837	0.9771	0.9755	0.9759	0.9796	0.9772	0.9794	0.9848	0.9774	0.9821
Guns	0.9709	0.9494	0.9755	0.0000	0.9699	0.9388	0.9784	0.9757	0.9747	0.9771	0.9814	0.9670	0.9720	0.9696	0.9724	0.9710	0.9734	0.9802	0.9726	0.9783
Mideast	0.9750	0.9727	0.9774	0.9699	0.0000	0.9520	0.9832	0.9833	0.9819	0.9828	0.9853	0.9781	0.9806	0.9774	0.9783	0.9795	0.9818	0.9865	0.9791	0.9827
Misc Politics	0.9699	0.9197	0.9740	0.9388	0.9520	0.0000	0.9799	0.9771	0.9762	0.9778	0.9828	0.9693	0.9741	0.9699	0.9713	0.9730	0.9762	0.9816	0.9743	0.9799
Graphics	0.9790	0.9778	0.9805	0.9784	0.9832	0.9799	0.0000	0.9660	0.9680	0.9650	0.9664	0.9758	0.9683	0.9746	0.9738	0.9733	0.9766	0.9750	0.9776	0.9804
IBM Hardware	0.9768	0.9748	0.9785	0.9757	0.9833	0.9771	0.9660	0.0000	0.9554	0.9587	0.9739	0.9743	0.9598	0.9738	0.9750	0.9674	0.9730	0.9680	0.9731	0.9769
Mac Hardware	0.9767	0.9747	0.9794	0.9747	0.9819	0.9762	0.9680	0.9554	0.0000	0.9674	0.9731	0.9730	0.9587	0.9726	0.9729	0.9677	0.9719	0.9675	0.9725	0.9770
Windows Misc	0.9786	0.9760	0.9808	0.9771	0.9828	0.9778	0.9650	0.9587	0.9674	0.0000	0.9715	0.9759	0.9686	0.9754	0.9762	0.9728	0.9767	0.9744	0.9766	0.9805
Windows X	0.9815	0.9813	0.9837	0.9814	0.9853	0.9828	0.9664	0.9739	0.9731	0.9715	0.0000	0.9776	0.9728	0.9791	0.9782	0.9777	0.9792	0.9801	0.9812	0.9829
Cryptography	0.9735	0.9709	0.9771	0.9670	0.9781	0.9693	0.9758	0.9743	0.9730	0.9759	0.9776	0.0000	0.9699	0.9738	0.9727	0.9732	0.9761	0.9796	0.9755	0.9807
Electronics	0.9737	0.9731	0.9755	0.9720	0.9806	0.9741	0.9683	0.9598	0.9587	0.9686	0.9728	0.9699	0.0000	0.9697	0.9694	0.9617	0.9681	0.9678	0.9706	0.9760
Medicine	0.9752	0.9698	0.9759	0.9696	0.9774	0.9699	0.9746	0.9738	0.9726	0.9754	0.9791	0.9738	0.9697	0.0000	0.9699	0.9721	0.9744	0.9776	0.9740	0.9794
Space	0.9762	0.9724	0.9796	0.9724	0.9783	0.9713	0.9738	0.9750	0.9729	0.9762	0.9782	0.9727	0.9694	0.9699	0.0000	0.9742	0.9771	0.9802	0.9770	0.9802
Autos	0.9760	0.9724	0.9772	0.9710	0.9795	0.9730	0.9733	0.9674	0.9677	0.9728	0.9777	0.9732	0.9617	0.9721	0.9742	0.0000	0.9643	0.9697	0.9696	0.9757
Motorcycles	0.9772	0.9754	0.9794	0.9734	0.9818	0.9762	0.9766	0.9730	0.9719	0.9767	0.9792	0.9761	0.9681	0.9744	0.9771	0.9643	0.0000	0.9772	0.9727	0.9786
For Sale	0.9822	0.9805	0.9848	0.9802	0.9865	0.9816	0.9750	0.9680	0.9675	0.9744	0.9801	0.9796	0.9678	0.9776	0.9802	0.9697	0.9772	0.0000	0.9771	0.9797
Baseball	0.9752	0.9740	0.9774	0.9726	0.9791	0.9743	0.9776	0.9731	0.9725	0.9766	0.9812	0.9755	0.9706	0.9740	0.9770	0.9696	0.9727	0.9771	0.0000	0.9680
Hockey	0.9797	0.9799	0.9821	0.9783	0.9827	0.9799	0.9804	0.9769	0.9770	0.9805	0.9829	0.9807	0.9760	0.9794	0.9802	0.9757	0.9786	0.9797	0.9680	0.0000

Table A.22: WGU Distance between Random Walk Samples (50%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0	30383	35456	35558	40094	38758	35718	31862	30862	42505	39011	35869	32129	37131	37161	32621	31484	32435	32060	35124
Religion	30383	0	37007	35757	41701	36647	37437	33525	32527	44078	40810	37466	33876	38504	38660	34172	33165	34134	33771	36949
Christian	35456	37007	0	36936	41348	40122	36890	33026	32084	43755	40249	37187	33295	38237	38471	33753	32680	33675	33258	36358
Guns	35558	35757	36936	0	40488	37162	36486	32596	31540	43181	39821	36203	32821	37515	37677	33093	32048	33115	32696	35832
Mideast	40094	41701	41348	40488	0	42220	41154	37420	36308	47997	44469	41289	37691	42373	42407	37953	36900	37885	37412	40464
Misc Politics	38758	36647	40122	37162	42220	0	39932	36010	34948	46563	43289	39637	36255	40809	40877	36515	35542	36561	36106	39282
Graphics	35718	37437	36890	36486	41154	39932	0	31566	30712	41743	38231	36431	32171	37479	37371	32831	31838	32357	32610	35562
IBM Hardware	31862	33525	33026	32596	37420	36010	31566	0	26404	37651	35133	32639	28053	33741	33779	28817	27940	28273	28646	31618
Mac Hardware	30862	32527	32084	31540	36308	34948	30712	26404	0	37333	34087	31565	27029	32671	32645	27855	26888	27271	27620	30624
Windows Misc	42505	44078	43755	43181	47997	46563	41743	37651	37333	0	45342	43210	38866	44304	44332	39532	38629	39074	39331	42413
Windows X	39011	40810	40249	39821	44469	43289	38231	35133	34087	45342	0	39652	35520	40916	40798	36216	35105	35796	35955	38869
Cryptography	35869	37466	37187	36203	41289	39637	36431	32639	31565	43210	39652	0	32814	37964	37836	33372	32353	33212	33021	36141
Electronics	32129	33876	33295	32821	37691	36255	32171	28053	27029	38866	35520	32814	0	33926	33858	28946	28123	28720	28961	32029
Medicine	37131	38504	38237	37515	42373	40809	37479	33741	32671	44304	40916	37964	33926	0	38744	34430	33375	34220	34055	37191
Space	37161	38660	38471	37677	42407	40877	37371	33779	32645	44332	40798	37836	33858	38744	0	34532	33515	34354	34217	37209
Autos	32621	34172	33753	33093	37953	36515	32831	28817	27855	39532	36216	33372	28946	34430	34532	0	28245	29172	29243	32349
Motorcycles	31484	33165	32680	32048	36900	35542	31838	27940	26888	38629	35105	32353	28123	33375	33515	28245	0	28395	28220	31316
For Sale	32435	34134	33675	33115	37885	36561	32357	28273	27271	39074	35796	33212	28720	34220	34354	29172	28395	0	29095	32017
Baseball	32060	33771	33258	32696	37412	36106	32610	28646	27620	39331	35955	33021	28961	34055	34217	29243	28220	29095	0	31356
Hockey	35124	36949	36358	35832	40464	39282	35562	31618	30624	42413	38869	36141	32029	37191	37209	32349	31316	32017	31356	0

Table A.23: UGU Distance between Random Walk Samples (50%)

	Atheism	Religion	Christian	Guns	Mideast	Misc Politics	Graphics	IBM Hardware	Mac Hardware	Windows Misc	Windows X	Cryptography	Electronics	Medicine	Space	Autos	Motorcycles	For Sale	Baseball	Hockey
Atheism	0.0000	0.6035	0.8105	0.7541	0.7678	0.7626	0.7761	0.7713	0.7868	0.7499	0.7855	0.7643	0.7606	0.7622	0.7632	0.7689	0.7709	0.8103	0.7831	0.7914
Religion	0.6035	0.0000	0.8004	0.7395	0.7836	0.6918	0.7911	0.7775	0.7854	0.7591	0.7910	0.7693	0.7750	0.7718	0.7785	0.7805	0.7837	0.8113	0.7950	0.7978
Christian	0.8105	0.8004	0.0000	0.8141	0.8211	0.8178	0.8265	0.8274	0.8298	0.8107	0.8229	0.8183	0.8138	0.8097	0.8143	0.8230	0.8260	0.8368	0.8337	0.8315
Guns	0.7541	0.7395	0.8141	0.0000	0.7688	0.6922	0.7734	0.7715	0.7752	0.7482	0.7830	0.7481	0.7614	0.7546	0.7584	0.7572	0.7688	0.7975	0.7822	0.7852
Mideast	0.7678	0.7836	0.8211	0.7688	0.0000	0.8014	0.8609	0.8668	0.8617	0.8443	0.8635	0.8598	0.8531	0.8472	0.8542	0.8622	0.8667	0.8775	0.8701	0.8726
Misc Politics	0.7626	0.6918	0.8178	0.6922	0.8014	0.0000	0.8185	0.8144	0.8159	0.7903	0.8210	0.7956	0.8105	0.7953	0.8017	0.7994	0.8150	0.8320	0.8175	0.8242
Graphics	0.7761	0.7911	0.8265	0.7734	0.8609	0.8185	0.0000	0.7773	0.7851	0.7318	0.7642	0.7823	0.7786	0.7885	0.7811	0.7976	0.8008	0.8070	0.8011	0.8118
IBM Hardware	0.7713	0.7775	0.8274	0.7715	0.8668	0.8144	0.7773	0.0000	0.7102	0.6616	0.7296	0.7362	0.7082	0.7308	0.7256	0.7184	0.7276	0.7323	0.7314	0.7526
Mac Hardware	0.7868	0.7854	0.8298	0.7752	0.8617	0.8159	0.7851	0.7102	0.0000	0.6695	0.7107	0.7136	0.6927	0.7124	0.7063	0.7090	0.7198	0.7167	0.7207	0.7294
Windows Misc	0.7499	0.7591	0.8107	0.7482	0.8443	0.7903	0.7318	0.6616	0.6695	0.0000	0.7746	0.7967	0.7833	0.7980	0.7956	0.7957	0.8023	0.8087	0.8111	0.8185
Windows X	0.7855	0.7910	0.8229	0.7830	0.8635	0.8210	0.7642	0.7296	0.7107	0.7746	0.0000	0.8315	0.8277	0.8410	0.8276	0.8390	0.8392	0.8362	0.8466	0.8497
Cryptography	0.7643	0.7693	0.8183	0.7481	0.8598	0.7956	0.7823	0.7362	0.7136	0.7967	0.8315	0.0000	0.8001	0.8002	0.8063	0.8135	0.8136	0.8339	0.8219	0.8332
Electronics	0.7606	0.7750	0.8138	0.7614	0.8531	0.8105	0.7786	0.7082	0.6927	0.7833	0.8277	0.8001	0.0000	0.6981	0.6902	0.6923	0.7026	0.7208	0.7243	0.7291
Medicine	0.7622	0.7718	0.8097	0.7546	0.8472	0.7953	0.7885	0.7308	0.7124	0.7980	0.8410	0.8002	0.6981	0.0000	0.7743	0.7863	0.7837	0.8135	0.8016	0.8059
Space	0.7632	0.7785	0.8143	0.7584	0.8542	0.8017	0.7811	0.7256	0.7063	0.7956	0.8276	0.8063	0.6902	0.7743	0.0000	0.7963	0.8056	0.8269	0.8157	0.8262
Autos	0.7689	0.7805	0.8230	0.7572	0.8622	0.7994	0.7976	0.7184	0.7090	0.7957	0.8390	0.8135	0.6923	0.7863	0.7963	0.0000	0.7007	0.7533	0.7255	0.7493
Motorcycles	0.7709	0.7837	0.8260	0.7688	0.8667	0.8150	0.8008	0.7276	0.7198	0.8023	0.8392	0.8136	0.7026	0.7837	0.8056	0.7007	0.0000	0.7720	0.7499	0.7663
For Sale	0.8103	0.8113	0.8368	0.7975	0.8775	0.8320	0.8070	0.7323	0.7167	0.8087	0.8362	0.8339	0.7208	0.8135	0.8269	0.7533	0.7720	0.0000	0.7670	0.7744
Baseball	0.7831	0.7950	0.8337	0.7822	0.8701	0.8175	0.8011	0.7314	0.7207	0.8111	0.8466	0.8219	0.7243	0.8016	0.8157	0.7255	0.7499	0.7670	0.0000	0.7252
Hockey	0.7914	0.7978	0.8315	0.7852	0.8726	0.8242	0.8118	0.7526	0.7294	0.8185	0.8497	0.8332	0.7291	0.8059	0.8262	0.7493	0.7663	0.7744	0.7252	0.0000

Table A.24: SID Distance between Random Walk Samples (50%)

Appendix B

Yahoo News! Articles Graph Distance Data

	Africa (W)	Canada (W)	Environment (Sc)	Mideast (W)	Soccer (Sp)	Animals (Sc)	Celebrity (E)	Europe (W)	Motor (Sp)	Software (T)	Apple (T)	Comm (T)	Fashion (E)	Movies (E)	Space (Sc)	Asia (W)	Crimes/Trials (US)	Football (Sp)	Music (E)	Tennis (S)	Australia (W)	Earnings (B)	Golf (Sp)	NCAA (Sp)	TV (E)	Baseball (Sp)	Economy (B)	Hockey (S)	Open source (T)	Weather (Sc)	Basketball (Sp)	Education (US)	Internet (T)	Personal (T)	Biotech (T)	Industry (E)	Latin (W)	Politics (US)	Books (E)	Energy (Sc)	Markets (B)	Religion (US)
Africa (W)	0.000	0.989	0.965	0.958	0.991	0.975	0.977	0.944	0.988	0.984	0.988	0.979	0.987	0.987	0.977	0.955	0.983	0.984	0.988	0.992	0.961	0.985	0.982	0.987	0.984	0.984	0.978	0.988	0.984	0.921	0.985	0.985	0.984	0.984	0.990	0.988	0.962	0.931	0.983	0.981	0.982	0.975
Canada (W)	0.989	0.000	0.987	0.992	0.982	0.990	0.994	0.989	0.996	0.994	0.992	0.990	0.994	0.994	0.991	0.989	0.992	0.994	0.994	0.991	0.987	0.987	0.993	0.993	0.994	0.993	0.982	0.990	0.992	0.983	0.991	0.993	0.991	0.990	0.986	0.993	0.992	0.989	0.995	0.989	0.987	0.990
Environment (Sc)	0.965	0.987	0.000	0.978	0.993	0.953	0.984	0.977	0.983	0.976	0.984	0.974	0.979	0.981	0.921	0.960	0.985	0.988	0.988	0.991	0.959	0.975	0.992	0.979	0.983	0.985	0.962	0.989	0.977	0.962	0.985	0.979	0.976	0.980	0.984	0.980	0.951	0.941	0.983	0.952	0.970	0.974
Mideast (W)	0.958	0.992	0.978	0.000	0.993	0.985	0.986	0.970	0.986	0.989	0.985	0.984	0.991	0.990	0.981	0.926	0.978	0.985	0.990	0.993	0.958	0.987	0.988	0.989	0.989	0.989	0.981	0.991	0.989	0.981	0.990	0.987	0.990	0.985	0.987	0.990	0.964	0.962	0.987	0.981	0.985	0.978
Soccer (Sp)	0.991	0.982	0.993	0.993	0.000	0.992	0.992	0.981	0.989	0.993	0.992	0.992	0.992	0.994	0.978	0.992	0.989	0.968	0.994	0.980	0.992	0.991	0.988	0.985	0.993	0.985	0.992	0.974	0.994	0.992	0.982	0.991	0.992	0.977	0.984	0.983	0.989	0.991	0.993	0.992	0.982	0.984
Animals (Sc)	0.975	0.990	0.953	0.985	0.992	0.000	0.986	0.985	0.987	0.984	0.986	0.980	0.987	0.988	0.971	0.954	0.978	0.986	0.986	0.993	0.943	0.986	0.992	0.988	0.989	0.985	0.979	0.990	0.987	0.971	0.984	0.985	0.987	0.984	0.940	0.990	0.978	0.978	0.985	0.982	0.984	0.985
Celebrity (E)	0.977	0.994	0.984	0.986	0.992	0.986	0.000	0.979	0.984	0.990	0.990	0.984	0.982	0.710	0.983	0.982	0.978	0.985	0.618	0.992	0.979	0.986	0.991	0.979	0.565	0.981	0.991	0.982	0.991	0.980	0.981	0.990	0.974	0.988	0.989	0.966	0.976	0.978	0.976	0.993	0.990	0.978
Europe (W)	0.944	0.989	0.977	0.970	0.981	0.985	0.979	0.000	0.986	0.988	0.994	0.985	0.942	0.990	0.984	0.965	0.937	0.986	0.990	0.991	0.966	0.986	0.992	0.987	0.988	0.986	0.986	0.984	0.988	0.981	0.984	0.989	0.985	0.985	0.993	0.988	0.969	0.975	0.987	0.985	0.990	0.981
Motor (Sp)	0.988	0.996	0.983	0.986	0.989	0.987	0.984	0.986	0.000	0.984	0.986	0.985	0.988	0.985	0.978	0.989	0.986	0.965	0.981	0.979	0.984	0.987	0.973	0.968	0.984	0.964	0.987	0.968	0.984	0.982	0.967	0.990	0.982	0.982	0.992	0.927	0.982	0.985	0.987	0.988	0.987	0.986
Software (T)	0.984	0.994	0.976	0.989	0.993	0.984	0.990	0.988	0.984	0.000	0.916	0.955	0.983	0.989	0.982	0.980	0.985	0.985	0.989	0.992	0.981	0.960	0.991	0.985	0.985	0.981	0.975	0.988	0.862	0.983	0.987	0.986	0.853	0.905	0.989	0.982	0.982	0.976	0.983	0.989	0.969	0.989
Apple (T)	0.988	0.992	0.984	0.985	0.992	0.986	0.990	0.994	0.986	0.916	0.000	0.948	0.991	0.989	0.987	0.986	0.989	0.989	0.990	0.993	0.984	0.950	0.993	0.987	0.987	0.986	0.973	0.991	0.944	0.986	0.986	0.985	0.968	0.900	0.990	0.981	0.985	0.985	0.985	0.989	0.970	0.988
Comm (T)	0.979	0.990	0.974	0.984	0.992	0.980	0.984	0.985	0.985	0.955	0.948	0.000	0.986	0.986	0.978	0.982	0.983	0.985	0.984	0.993	0.975	0.885	0.989	0.983	0.980	0.983	0.958	0.987	0.967	0.972	0.988	0.982	0.967	0.514	0.987	0.946	0.981	0.974	0.984	0.983	0.956	0.983
Fashion (E)	0.987	0.994	0.979	0.991	0.992	0.987	0.982	0.942	0.988	0.983	0.991	0.986	0.000	0.978	0.988	0.985	0.990	0.988	0.982	0.991	0.988	0.985	0.994	0.989	0.979	0.987	0.991	0.988	0.989	0.985	0.987	0.993	0.979	0.989	0.994	0.980	0.990	0.982	0.985	0.990	0.987	0.988
Movies (E)	0.987	0.994	0.981	0.990	0.994	0.988	0.710	0.990	0.985	0.989	0.989	0.986	0.978	0.000	0.986	0.983	0.982	0.987	0.818	0.993	0.985	0.987	0.992	0.980	0.683	0.980	0.993	0.986	0.990	0.981	0.985	0.987	0.977	0.988	0.992	0.856	0.982	0.981	0.977	0.991	0.991	0.976
Space (Sc)	0.977	0.991	0.921	0.981	0.978	0.971	0.983	0.984	0.978	0.982	0.987	0.978	0.988	0.986	0.000	0.970	0.885	0.964	0.984	0.989	0.971	0.987	0.988	0.978	0.988	0.984	0.980	0.986	0.983	0.970	0.983	0.986	0.981	0.954	0.988	0.962	0.980	0.974	0.988	0.980	0.950	0.953
Asia (W)	0.955	0.989	0.960	0.926	0.992	0.954	0.982	0.965	0.989	0.980	0.986	0.982	0.985	0.983	0.970	0.000	0.963	0.985	0.989	0.992	0.937	0.978	0.990	0.985	0.986	0.982	0.974	0.989	0.984	0.964	0.988	0.983	0.983	0.980	0.988	0.990	0.960	0.954	0.983	0.978	0.975	0.972
Crimes/Trials (US)	0.983	0.992	0.985	0.978	0.989	0.978	0.978	0.937	0.986	0.985	0.989	0.983	0.990	0.982	0.885	0.963	0.000	0.977	0.957	0.990	0.976	0.985	0.989	0.982	0.985	0.982	0.984	0.987	0.989	0.948	0.985	0.983	0.991	0.985	0.989	0.990	0.975	0.979	0.985	0.993	0.987	0.979
Football (Sp)	0.984	0.994	0.988	0.985	0.968	0.986	0.985	0.986	0.965	0.985	0.989	0.985	0.988	0.987	0.964	0.985	0.977	0.000	0.984	0.984	0.986	0.983	0.978	0.959	0.984	0.947	0.984	0.938	0.986	0.977	0.954	0.988	0.987	0.970	0.992	0.968	0.983	0.988	0.988	0.989	0.969	0.970
Music (E)	0.988	0.994	0.988	0.990	0.994	0.986	0.618	0.990	0.981	0.989	0.990	0.984	0.982	0.818	0.984	0.989	0.957	0.984	0.000	0.991	0.989	0.988	0.990	0.976	0.730	0.979	0.991	0.985	0.989	0.980	0.983	0.989	0.917	0.988	0.990	0.951	0.986	0.983	0.982	0.994	0.991	0.979
Tennis (S)	0.992	0.991	0.991	0.993	0.980	0.993	0.992	0.991	0.979	0.992	0.993	0.993	0.991	0.993	0.989	0.992	0.990	0.984	0.991	0.000	0.991	0.989	0.966	0.977	0.991	0.979	0.993	0.978	0.992	0.990	0.978	0.991	0.992	0.994	0.993	0.992	0.990	0.993	0.994	0.992	0.993	0.992
Australia (W)	0.961	0.987	0.959	0.958	0.992	0.943	0.979	0.966	0.984	0.981	0.984	0.975	0.988	0.985	0.971	0.937	0.976	0.986	0.989	0.991	0.000	0.975	0.988	0.980	0.985	0.984	0.973	0.989	0.984	0.968	0.982	0.982	0.975	0.979	0.991	0.988	0.962	0.952	0.980	0.972	0.975	0.975
Earnings (B)	0.985	0.987	0.975	0.987	0.991	0.986	0.986	0.986	0.987	0.960	0.950	0.885	0.985	0.987	0.987	0.978	0.985	0.983	0.988	0.989	0.975	0.000	0.990	0.988	0.982	0.984	0.932	0.983	0.967	0.978	0.985	0.980	0.970	0.937	0.988	0.977	0.979	0.977	0.987	0.984	0.898	0.985
Golf (Sp)	0.982	0.993	0.992	0.988	0.988	0.992	0.991	0.992	0.973	0.991	0.993	0.989	0.994	0.992	0.988	0.990	0.989	0.978	0.990	0.966	0.988	0.990	0.000	0.980	0.992	0.981	0.991	0.975	0.990	0.988	0.979	0.988	0.988	0.990	0.994	0.990	0.986	0.992	0.991	0.991	0.990	0.992
NCAA (Sp)	0.987	0.993	0.979	0.989	0.985	0.988	0.979	0.987	0.968	0.985	0.987	0.983	0.989	0.980	0.978	0.985	0.982	0.959	0.976	0.977	0.980	0.988	0.980	0.000	0.978	0.959	0.988	0.955	0.984	0.972	0.922	0.983	0.982	0.984	0.992	0.982	0.983	0.980	0.986	0.989	0.989	0.981
TV (E)	0.984	0.994	0.983	0.989	0.993	0.989	0.565	0.988	0.984	0.985	0.987	0.980	0.979	0.683	0.988	0.986	0.985	0.984	0.730	0.991	0.985	0.982	0.992	0.978	0.000	0.982	0.988	0.984	0.987	0.981	0.981	0.988	0.881	0.979	0.992	0.955	0.981	0.976	0.981	0.992	0.986	0.980
Baseball (Sp)	0.984	0.993	0.985	0.989	0.985	0.985	0.981	0.986	0.964	0.981	0.986	0.983	0.987	0.980	0.984	0.982	0.982	0.947	0.979	0.979	0.984	0.984	0.981	0.959	0.982	0.000	0.987	0.956	0.986	0.979	0.947	0.986	0.984	0.984	0.994	0.980	0.985	0.979	0.986	0.993	0.985	0.980
Economy (B)	0.978	0.982	0.962	0.981	0.992	0.979	0.991	0.986	0.987	0.975	0.973	0.958	0.991	0.993	0.980	0.974	0.984	0.984	0.991	0.993	0.973	0.932	0.991	0.988	0.988	0.987	0.000	0.986	0.983	0.976	0.984	0.980	0.982	0.966	0.987	0.984	0.982	0.938	0.986	0.969	0.869	0.984
Hockey (S)	0.988	0.990	0.989	0.991	0.974	0.990	0.982	0.984	0.968	0.988	0.991	0.987	0.988	0.986	0.986	0.989	0.987	0.938	0.985	0.978	0.989	0.983	0.975	0.955	0.984	0.956	0.986	0.000	0.990	0.985	0.934	0.991	0.987	0.985	0.995	0.985	0.987	0.990	0.987	0.995	0.989	0.989
Open source (T)	0.984	0.992	0.977	0.989	0.994	0.987	0.991	0.988	0.984	0.862	0.944	0.967	0.989	0.990	0.983	0.984	0.989	0.986	0.989	0.992	0.984	0.967	0.990	0.984	0.987	0.986	0.983	0.990	0.000	0.983	0.988	0.987	0.970	0.967	0.991	0.984	0.985	0.982	0.984	0.988	0.980	0.985
Weather (Sc)	0.921	0.983	0.962	0.981	0.992	0.971	0.980	0.981	0.982	0.983	0.986	0.972	0.985	0.981	0.970	0.964	0.948	0.977	0.980	0.990	0.968	0.978	0.988	0.972	0.981	0.979	0.976	0.985	0.983	0.000	0.969	0.943	0.982	0.977	0.990	0.983	0.967	0.865	0.982	0.979	0.980	0.978
Basketball (Sp)	0.985	0.991	0.985	0.990	0.982	0.984	0.981	0.984	0.967	0.987	0.986	0.988	0.987	0.985	0.983	0.988	0.985	0.954	0.983	0.978	0.982	0.985	0.979	0.922	0.981	0.947	0.984	0.934	0.988	0.969	0.000	0.988	0.981	0.986	0.993	0.984	0.985	0.984	0.988	0.992	0.986	0.988
Education (US)	0.985	0.993	0.979	0.987	0.991	0.985	0.990	0.989	0.990	0.986	0.985	0.982	0.993	0.987	0.986	0.983	0.983	0.988	0.989	0.991	0.982	0.980	0.988	0.983	0.988	0.986	0.980	0.991	0.987	0.943	0.988	0.000	0.989	0.985	0.987	0.989	0.985	0.944	0.982	0.982	0.987	0.981
Internet (T)	0.984	0.991	0.976	0.990	0.992	0.987	0.974	0.985	0.982	0.853	0.968	0.967	0.979	0.977	0.981	0.983	0.991	0.987	0.917	0.992	0.975	0.970	0.988	0.982	0.881	0.984	0.982	0.987	0.970	0.982	0.981	0.989	0.000	0.827	0.995	0.885	0.989	0.977	0.985	0.988	0.975	0.983
Personal (T)	0.984	0.990	0.980	0.985	0.977	0.984	0.988	0.985	0.982	0.905	0.900	0.514	0.989	0.988	0.954	0.980	0.985	0.970	0.988	0.994	0.979	0.937	0.990	0.984	0.979	0.984	0.966	0.985	0.967	0.977	0.986	0.985	0.827	0.000	0.987	0.955	0.983	0.980	0.986	0.983	0.934	0.965
Biotech (T)	0.990	0.986	0.984	0.987	0.984	0.940	0.989	0.993	0.992	0.989	0.990	0.987	0.994	0.992	0.988	0.988	0.989	0.992	0.990	0.993	0.991	0.988	0.994	0.992	0.992	0.994	0.987	0.995	0.991	0.990	0.993	0.987	0.995	0.987	0.000	0.994	0.987	0.986	0.990	0.979	0.991	0.986
Industry (E)	0.988	0.993	0.980	0.990	0.983	0.990	0.966	0.988	0.927	0.982	0.981	0.946	0.980	0.856	0.962	0.990	0.990	0.968	0.951	0.992	0.988	0.977	0.990	0.982	0.955	0.980	0.984	0.985	0.984	0.983	0.984	0.989	0.885	0.955	0.994	0.000	0.987	0.984	0.984	0.990	0.957	0.941
Latin (W)	0.962	0.992	0.951	0.964	0.989	0.978	0.976	0.969	0.982	0.982	0.985	0.981	0.990	0.982	0.980	0.960	0.975	0.983	0.986	0.990	0.962	0.979	0.986	0.983	0.981	0.985	0.982	0.987	0.985	0.967	0.985	0.985	0.989	0.983	0.987	0.987	0.000	0.966	0.951	0.979	0.985	0.978
Politics (US)	0.931	0.989	0.941	0.962	0.991	0.978	0.978	0.975	0.985	0.976	0.985	0.974	0.982	0.981	0.974	0.954	0.979	0.988	0.983	0.993	0.952	0.977	0.992	0.980	0.976	0.979	0.938	0.990	0.982	0.865	0.984	0.944	0.977	0.980	0.986	0.984	0.966	0.000	0.978	0.977	0.971	0.951
Books (E)	0.983	0.995	0.983	0.987	0.993	0.985	0.976	0.987	0.987	0.983	0.985	0.984	0.985	0.977	0.988	0.983	0.985	0.988	0.982	0.994	0.980	0.987	0.991	0.986	0.981	0.986	0.986	0.987	0.984	0.982	0.988	0.982	0.985	0.986	0.990	0.984	0.951	0.978	0.000	0.989	0.989	0.980
Energy (Sc)	0.981	0.989	0.952	0.981	0.992	0.982	0.993	0.985	0.988	0.989	0.989	0.983	0.990	0.991	0.980	0.978	0.993	0.989	0.994	0.992	0.972	0.984	0.991	0.989	0.992	0.993	0.969	0.995	0.988	0.979	0.992	0.982	0.988	0.983	0.979	0.990	0.979	0.977	0.989	0.000	0.984	0.983
Markets (B)	0.982	0.987	0.970	0.985	0.982	0.984	0.990	0.990	0.987	0.969	0.970	0.956	0.987	0.991	0.950	0.975	0.987	0.969	0.991	0.993	0.975	0.898	0.990	0.989	0.986	0.985	0.869	0.989	0.980	0.980	0.986	0.987	0.975	0.934	0.991	0.957	0.985	0.971	0.989	0.984	0.000	0.961
Religion (US)	0.975	0.990	0.974	0.978	0.984	0.985	0.978	0.981	0.986	0.989	0.988	0.983	0.988	0.976	0.953	0.972	0.979	0.970	0.979	0.992	0.975	0.985	0.992	0.981	0.980	0.980	0.984	0.989	0.985	0.978	0.988	0.981	0.983	0.965	0.986	0.941	0.978	0.951	0.980	0.983	0.961	0.000

Table B.1: mcs Distance between Yahoo! News Graphs

	Africa (W)	Canada (W)	Environment (Sc)	Mideast (W)	Soccer (Sp)	Animals (Sc)	Celebrity (E)	Europe (W)	Motor (Sp)	Software (T)	Apple (T)	Comm (T)	Fashion (E)	Movies (E)	Space (Sc)	Asia (W)	Crimes/Trials (US)	Football (Sp)	Music (E)	Tennis (S)	Australia (W)	Earnings (B)	Golf (Sp)	NCAA (Sp)	TV (E)	Baseball (Sp)	Economy (B)	Hockey (S)	Open source (T)	Weather (Sc)	Basketball (Sp)	Education (US)	Internet (T)	Personal (T)	Biotech (T)	Industry (E)	Latin (W)	Politics (US)	Books (E)	Energy (Sc)	Markets (B)	Religion (US)
Africa (W)	0.000	0.991	0.979	0.971	0.992	0.986	0.985	0.966	0.993	0.991	0.993	0.987	0.992	0.991	0.986	0.972	0.989	0.989	0.993	0.994	0.975	0.991	0.987	0.992	0.990	0.991	0.986	0.992	0.991	0.952	0.991	0.990	0.991	0.990	0.992	0.992	0.977	0.957	0.990	0.986	0.990	0.982
Canada (W)	0.991	0.000	0.989	0.994	0.990	0.992	0.995	0.991	0.997	0.995	0.994	0.992	0.995	0.995	0.992	0.991	0.994	0.995	0.995	0.995	0.989	0.990	0.995	0.995	0.995	0.994	0.986	0.992	0.994	0.986	0.993	0.995	0.993	0.992	0.992	0.994	0.993	0.990	0.996	0.993	0.989	0.992
Environment (Sc)	0.979	0.989	0.000	0.985	0.994	0.972	0.990	0.986	0.989	0.986	0.990	0.984	0.988	0.987	0.951	0.975	0.990	0.992	0.993	0.993	0.973	0.984	0.994	0.988	0.990	0.991	0.975	0.993	0.986	0.976	0.991	0.985	0.986	0.988	0.987	0.986	0.969	0.964	0.991	0.962	0.982	0.982
Mideast (W)	0.971	0.994	0.985	0.000	0.995	0.990	0.990	0.980	0.991	0.992	0.991	0.990	0.994	0.992	0.988	0.950	0.987	0.992	0.993	0.995	0.973	0.992	0.993	0.992	0.993	0.992	0.989	0.995	0.993	0.987	0.994	0.993	0.993	0.991	0.991	0.993	0.977	0.973	0.991	0.988	0.990	0.983
Soccer (Sp)	0.992	0.990	0.994	0.995	0.000	0.993	0.993	0.985	0.992	0.995	0.994	0.993	0.994	0.995	0.983	0.994	0.992	0.976	0.995	0.988	0.994	0.993	0.992	0.988	0.994	0.988	0.994	0.980	0.995	0.994	0.985	0.994	0.994	0.982	0.991	0.985	0.991	0.993	0.995	0.995	0.986	0.986
Animals (Sc)	0.986	0.992	0.972	0.990	0.993	0.000	0.991	0.991	0.992	0.991	0.991	0.989	0.992	0.992	0.983	0.973	0.986	0.991	0.991	0.995	0.964	0.992	0.994	0.993	0.993	0.991	0.987	0.994	0.993	0.983	0.991	0.990	0.992	0.990	0.952	0.993	0.987	0.986	0.991	0.986	0.991	0.989
Celebrity (E)	0.985	0.995	0.990	0.990	0.993	0.991	0.000	0.986	0.989	0.993	0.993	0.990	0.989	0.801	0.989	0.988	0.983	0.989	0.737	0.993	0.985	0.991	0.993	0.987	0.688	0.988	0.994	0.987	0.994	0.987	0.988	0.992	0.983	0.992	0.991	0.978	0.983	0.987	0.984	0.994	0.994	0.986
Europe (W)	0.966	0.991	0.986	0.980	0.985	0.991	0.986	0.000	0.992	0.993	0.996	0.991	0.964	0.993	0.991	0.979	0.958	0.991	0.994	0.994	0.979	0.992	0.994	0.992	0.992	0.992	0.991	0.990	0.993	0.989	0.991	0.993	0.991	0.991	0.994	0.992	0.981	0.984	0.993	0.989	0.994	0.986
Motor (Sp)	0.993	0.997	0.989	0.991	0.992	0.992	0.989	0.992	0.000	0.990	0.992	0.991	0.992	0.989	0.987	0.994	0.991	0.978	0.987	0.984	0.991	0.993	0.982	0.980	0.990	0.977	0.992	0.980	0.990	0.989	0.980	0.994	0.989	0.990	0.994	0.944	0.990	0.990	0.992	0.992	0.992	0.990
Software (T)	0.991	0.995	0.986	0.992	0.995	0.991	0.993	0.993	0.990	0.000	0.944	0.972	0.990	0.993	0.990	0.988	0.990	0.990	0.993	0.994	0.988	0.975	0.994	0.991	0.991	0.990	0.984	0.992	0.913	0.990	0.993	0.990	0.905	0.939	0.991	0.987	0.989	0.985	0.990	0.992	0.982	0.993
Apple (T)	0.993	0.994	0.990	0.991	0.994	0.991	0.993	0.996	0.992	0.944	0.000	0.967	0.994	0.992	0.992	0.992	0.993	0.994	0.993	0.995	0.991	0.970	0.996	0.992	0.991	0.991	0.985	0.995	0.963	0.992	0.992	0.991	0.979	0.938	0.993	0.986	0.992	0.990	0.991	0.992	0.982	0.991
Comm (T)	0.987	0.992	0.984	0.990	0.993	0.989	0.990	0.991	0.991	0.972	0.967	0.000	0.992	0.990	0.988	0.990	0.989	0.990	0.990	0.995	0.984	0.927	0.992	0.990	0.987	0.990	0.973	0.992	0.980	0.984	0.993	0.988	0.979	0.638	0.990	0.960	0.989	0.984	0.991	0.988	0.974	0.988
Fashion (E)	0.992	0.995	0.988	0.994	0.994	0.992	0.989	0.964	0.992	0.990	0.994	0.992	0.000	0.986	0.993	0.991	0.993	0.992	0.989	0.993	0.992	0.991	0.996	0.994	0.988	0.992	0.994	0.992	0.993	0.991	0.992	0.995	0.988	0.993	0.995	0.987	0.994	0.990	0.991	0.993	0.992	0.992
Movies (E)	0.991	0.995	0.987	0.992	0.995	0.992	0.801	0.993	0.989	0.993	0.992	0.990	0.986	0.000	0.990	0.988	0.986	0.990	0.879	0.994	0.989	0.991	0.993	0.987	0.773	0.987	0.995	0.990	0.993	0.987	0.990	0.990	0.985	0.992	0.994	0.907	0.987	0.987	0.985	0.992	0.994	0.986
Space (Sc)	0.986	0.992	0.951	0.988	0.983	0.983	0.989	0.991	0.987	0.990	0.992	0.988	0.993	0.990	0.000	0.983	0.923	0.977	0.989	0.991	0.982	0.992	0.992	0.987	0.993	0.991	0.988	0.991	0.990	0.982	0.991	0.991	0.988	0.973	0.991	0.973	0.988	0.984	0.993	0.985	0.971	0.967
Asia (W)	0.972	0.991	0.975	0.950	0.994	0.973	0.988	0.979	0.994	0.988	0.992	0.990	0.991	0.988	0.983	0.000	0.976	0.991	0.993	0.994	0.960	0.987	0.993	0.991	0.991	0.989	0.984	0.993	0.991	0.979	0.993	0.989	0.989	0.988	0.991	0.993	0.976	0.970	0.990	0.984	0.985	0.980
Crimes/Trials (US)	0.989	0.994	0.990	0.987	0.992	0.986	0.983	0.958	0.991	0.990	0.993	0.989	0.993	0.986	0.923	0.976	0.000	0.987	0.968	0.994	0.985	0.991	0.994	0.988	0.989	0.988	0.991	0.993	0.993	0.965	0.990	0.991	0.993	0.991	0.992	0.992	0.984	0.985	0.990	0.995	0.991	0.984
Football (Sp)	0.989	0.995	0.992	0.992	0.976	0.991	0.989	0.991	0.978	0.990	0.994	0.990	0.992	0.990	0.977	0.991	0.987	0.000	0.988	0.989	0.991	0.990	0.987	0.971	0.988	0.963	0.991	0.962	0.991	0.985	0.970	0.993	0.991	0.982	0.994	0.975	0.990	0.991	0.992	0.993	0.980	0.977
Music (E)	0.993	0.995	0.993	0.993	0.995	0.991	0.737	0.994	0.987	0.993	0.993	0.990	0.989	0.879	0.989	0.993	0.968	0.988	0.000	0.993	0.992	0.992	0.992	0.985	0.823	0.986	0.994	0.990	0.993	0.987	0.989	0.992	0.948	0.992	0.992	0.968	0.990	0.989	0.989	0.995	0.994	0.987
Tennis (S)	0.994	0.995	0.993	0.995	0.988	0.995	0.993	0.994	0.984	0.994	0.995	0.995	0.993	0.994	0.991	0.994	0.994	0.989	0.993	0.000	0.994	0.992	0.977	0.982	0.993	0.983	0.995	0.984	0.994	0.992	0.983	0.994	0.994	0.995	0.996	0.993	0.993	0.995	0.995	0.995	0.994	0.993
Australia (W)	0.975	0.989	0.973	0.973	0.994	0.964	0.985	0.979	0.991	0.988	0.991	0.984	0.992	0.989	0.982	0.960	0.985	0.991	0.992	0.994	0.000	0.985	0.992	0.987	0.990	0.990	0.984	0.994	0.990	0.980	0.989	0.989	0.984	0.988	0.993	0.991	0.977	0.967	0.987	0.980	0.985	0.981
Earnings (B)	0.991	0.990	0.984	0.992	0.993	0.992	0.991	0.992	0.993	0.975	0.970	0.927	0.991	0.991	0.992	0.987	0.991	0.990	0.992	0.992	0.985	0.000	0.994	0.992	0.988	0.990	0.958	0.990	0.980	0.987	0.991	0.987	0.980	0.963	0.991	0.982	0.988	0.985	0.992	0.988	0.934	0.989
Golf (Sp)	0.987	0.995	0.994	0.993	0.992	0.994	0.993	0.994	0.982	0.994	0.996	0.992	0.996	0.993	0.992	0.993	0.994	0.987	0.992	0.977	0.992	0.994	0.000	0.985	0.994	0.986	0.995	0.983	0.993	0.992	0.985	0.993	0.991	0.994	0.996	0.992	0.991	0.994	0.993	0.995	0.993	0.994
NCAA (Sp)	0.992	0.995	0.988	0.992	0.988	0.993	0.987	0.992	0.980	0.991	0.992	0.990	0.994	0.987	0.987	0.991	0.988	0.971	0.985	0.982	0.987	0.992	0.985	0.000	0.987	0.975	0.992	0.970	0.991	0.983	0.951	0.988	0.990	0.990	0.994	0.987	0.990	0.988	0.992	0.992	0.994	0.987
TV (E)	0.990	0.995	0.990	0.993	0.994	0.993	0.688	0.992	0.990	0.991	0.991	0.987	0.988	0.773	0.993	0.991	0.989	0.988	0.823	0.993	0.990	0.988	0.994	0.987	0.000	0.989	0.992	0.989	0.992	0.988	0.988	0.991	0.927	0.986	0.993	0.969	0.988	0.986	0.988	0.994	0.991	0.987
Baseball (Sp)	0.991	0.994	0.991	0.992	0.988	0.991	0.988	0.992	0.977	0.990	0.991	0.990	0.992	0.987	0.991	0.989	0.988	0.963	0.986	0.983	0.990	0.990	0.986	0.975	0.989	0.000	0.992	0.971	0.992	0.988	0.968	0.991	0.990	0.991	0.995	0.986	0.991	0.988	0.992	0.994	0.992	0.986
Economy (B)	0.986	0.986	0.975	0.989	0.994	0.987	0.994	0.991	0.992	0.984	0.985	0.973	0.994	0.995	0.988	0.984	0.991	0.991	0.994	0.995	0.984	0.958	0.995	0.992	0.992	0.992	0.000	0.992	0.989	0.985	0.990	0.988	0.988	0.980	0.990	0.988	0.989	0.958	0.991	0.978	0.911	0.988
Hockey (S)	0.992	0.992	0.993	0.995	0.980	0.994	0.987	0.990	0.980	0.992	0.995	0.992	0.992	0.990	0.991	0.993	0.993	0.962	0.990	0.984	0.994	0.990	0.983	0.970	0.989	0.971	0.992	0.000	0.994	0.991	0.957	0.995	0.991	0.991	0.997	0.989	0.992	0.994	0.992	0.997	0.993	0.992
Open source (T)	0.991	0.994	0.986	0.993	0.995	0.993	0.994	0.993	0.990	0.913	0.963	0.980	0.993	0.993	0.990	0.991	0.993	0.991	0.993	0.994	0.990	0.980	0.993	0.991	0.992	0.992	0.989	0.994	0.000	0.990	0.993	0.991	0.981	0.980	0.993	0.988	0.991	0.989	0.991	0.991	0.988	0.989
Weather (Sc)	0.952	0.986	0.976	0.987	0.994	0.983	0.987	0.989	0.989	0.990	0.992	0.984	0.991	0.987	0.982	0.979	0.965	0.985	0.987	0.992	0.980	0.987	0.992	0.983	0.988	0.988	0.985	0.991	0.990	0.000	0.983	0.962	0.989	0.986	0.992	0.987	0.980	0.912	0.989	0.984	0.989	0.984
Basketball (Sp)	0.991	0.993	0.991	0.994	0.985	0.991	0.988	0.991	0.980	0.993	0.992	0.993	0.992	0.990	0.991	0.993	0.990	0.970	0.989	0.983	0.989	0.991	0.985	0.951	0.988	0.968	0.990	0.957	0.993	0.983	0.000	0.992	0.988	0.992	0.995	0.988	0.991	0.990	0.993	0.994	0.992	0.992
Education (US)	0.990	0.995	0.985	0.993	0.994	0.990	0.992	0.993	0.994	0.990	0.991	0.988	0.995	0.990	0.991	0.989	0.991	0.993	0.992	0.994	0.989	0.987	0.993	0.988	0.991	0.991	0.988	0.995	0.991	0.962	0.992	0.000	0.992	0.991	0.991	0.990	0.958	0.988	0.988	0.991	0.985	0.989
Internet (T)	0.991	0.993	0.986	0.993	0.994	0.992	0.983	0.991	0.989	0.905	0.979	0.979	0.988	0.985	0.988	0.989	0.993	0.991	0.948	0.994	0.984	0.980	0.991	0.990	0.927	0.990	0.988	0.991	0.981	0.989	0.988	0.992	0.000	0.880	0.996	0.917	0.993	0.986	0.991	0.991	0.985	0.989
Personal (T)	0.990	0.992	0.988	0.991	0.982	0.990	0.992	0.991	0.990	0.939	0.938	0.638	0.993	0.992	0.973	0.988	0.991	0.982	0.992	0.995	0.988	0.963	0.994	0.990	0.986	0.991	0.980	0.991	0.980	0.986	0.992	0.991	0.880	0.000	0.990	0.966	0.991	0.987	0.991	0.988	0.959	0.975
Biotech (T)	0.992	0.992	0.987	0.991	0.991	0.952	0.991	0.994	0.994	0.991	0.993	0.990	0.995	0.994	0.991	0.991	0.992	0.994	0.992	0.996	0.993	0.991	0.996	0.994	0.993	0.995	0.990	0.997	0.993	0.992	0.995	0.991	0.996	0.990	0.000	0.995	0.991	0.989	0.992	0.988	0.993	0.988
Industry (E)	0.992	0.994	0.986	0.993	0.985	0.993	0.978	0.992	0.944	0.987	0.986	0.960	0.987	0.907	0.973	0.993	0.992	0.975	0.968	0.993	0.991	0.982	0.992	0.987	0.969	0.986	0.988	0.989	0.988	0.987	0.988	0.991	0.917	0.966	0.995	0.000	0.991	0.989	0.988	0.992	0.970	0.965
Latin (W)	0.977	0.993	0.969	0.977	0.991	0.987	0.983	0.981	0.990	0.989	0.992	0.989	0.994	0.987	0.988	0.976	0.984	0.990	0.990	0.993	0.977	0.988	0.991	0.990	0.988	0.991	0.989	0.992	0.991	0.980	0.991	0.990	0.993	0.991	0.991	0.991	0.000	0.978	0.969	0.985	0.991	0.984
Politics (US)	0.957	0.990	0.964	0.973	0.993	0.986	0.987	0.984	0.990	0.985	0.990	0.984	0.990	0.987	0.984	0.970	0.985	0.991	0.989	0.995	0.967	0.985	0.994	0.988	0.986	0.988	0.958	0.994	0.989	0.912	0.990	0.958	0.986	0.987	0.989	0.989	0.978	0.000	0.987	0.982	0.982	0.966
Books (E)	0.990	0.996	0.991	0.991	0.995	0.991	0.984	0.993	0.992	0.990	0.991	0.991	0.991	0.985	0.993	0.990	0.990	0.992	0.989	0.995	0.987	0.992	0.993	0.992	0.988	0.992	0.991	0.992	0.991	0.989	0.993	0.988	0.991	0.991	0.992	0.988	0.969	0.987	0.000	0.992	0.994	0.986
Energy (Sc)	0.986	0.993	0.962	0.988	0.995	0.986	0.994	0.989	0.992	0.992	0.992	0.988	0.993	0.992	0.985	0.984	0.995	0.993	0.995	0.995	0.980	0.988	0.995	0.992	0.994	0.994	0.978	0.997	0.991	0.984	0.994	0.988	0.991	0.988	0.988	0.992	0.985	0.982	0.992	0.000	0.988	0.985
Markets (B)	0.990	0.989	0.982	0.990	0.986	0.991	0.994	0.994	0.992	0.982	0.982	0.974	0.992	0.994	0.971	0.985	0.991	0.980	0.994	0.994	0.985	0.934	0.993	0.994	0.991	0.992	0.911	0.993	0.988	0.989	0.992	0.991	0.985	0.959	0.993	0.970	0.991	0.982	0.994	0.988	0.000	0.973
Religion (US)	0.982	0.992	0.982	0.983	0.986	0.989	0.986	0.986	0.990	0.993	0.991	0.988	0.992	0.986	0.967	0.980	0.984	0.977	0.987	0.993	0.981	0.989	0.994	0.987	0.987	0.986	0.988	0.992	0.989	0.984	0.992	0.985	0.989	0.975	0.988	0.965	0.984	0.966	0.986	0.985	0.973	0.000

Table B.2: MCS Distance between Yahoo! News Graphs

	Africa (W)	Canada (W)	Environment (Sc)	Mideast (W)	Soccer (Sp)	Animals (Sc)	Celebrity (E)	Europe (W)	Motor (Sp)	Software (T)	Apple (T)	Comm (T)	Fashion (E)	Movies (E)	Space (Sc)	Asia (W)	Crimes/Trials (US)	Football (Sp)	Music (E)	Tennis (S)	Australia (W)	Earnings (B)	Golf (Sp)	NCAA (Sp)	TV (E)	Baseball (Sp)	Economy (B)	Hockey (S)	Open source (T)	Weather (Sc)	Basketball (Sp)	Education (US)	Internet (T)	Personal (T)	Biotech (T)	Industry (E)	Latin (W)	Politics (US)	Books (E)	Energy (Sc)	Markets (B)	Religion (US)
Africa (W)	0.000	0.995	0.978	0.981	0.997	0.986	0.986	0.962	0.997	0.992	0.994	0.989	0.992	0.991	0.991	0.976	0.987	0.993	0.993	0.996	0.978	0.992	0.992	0.997	0.995	0.993	0.990	0.995	0.997	0.952	0.993	0.994	0.992	0.991	0.995	0.994	0.985	0.949	0.993	0.989	0.989	0.982
Canada (W)	0.995	0.000	0.985	0.995	0.993	0.985	0.991	0.988	0.998	0.994	0.989	0.988	0.998	0.988	0.984	0.991	0.990	0.992	0.991	0.995	0.984	0.984	0.995	0.993	0.991	0.988	0.985	0.990	0.989	0.984	0.988	0.995	0.981	0.991	0.992	0.988	0.992	0.981	0.993	0.996	0.980	0.976
Environment (Sc)	0.978	0.985	0.000	0.989	0.996	0.968	0.992	0.984	0.993	0.987	0.991	0.988	0.987	0.989	0.942	0.980	0.990	0.991	0.994	0.996	0.969	0.987	0.995	0.995	0.991	0.992	0.981	0.995	0.992	0.977	0.988	0.989	0.981	0.988	0.992	0.984	0.970	0.961	0.988	0.957	0.984	0.971
Mideast (W)	0.981	0.995	0.989	0.000	0.997	0.992	0.992	0.988	0.995	0.994	0.995	0.994	0.997	0.996	0.991	0.951	0.992	0.995	0.995	0.997	0.978	0.995	0.995	0.995	0.992	0.996	0.991	0.997	0.994	0.994	0.997	0.995	0.994	0.993	0.994	0.994	0.978	0.977	0.992	0.989	0.997	0.983
Soccer (Sp)	0.997	0.993	0.996	0.997	0.000	0.993	0.988	0.983	0.986	0.994	0.995	0.994	0.991	0.990	0.969	0.994	0.989	0.964	0.994	0.987	0.997	0.988	0.994	0.992	0.989	0.984	0.995	0.981	0.993	0.995	0.985	0.995	0.993	0.964	0.994	0.946	0.993	0.994	0.994	0.998	0.972	0.952
Animals (Sc)	0.986	0.985	0.968	0.992	0.993	0.000	0.993	0.993	0.995	0.988	0.994	0.990	0.993	0.990	0.981	0.976	0.989	0.993	0.993	0.998	0.951	0.993	0.994	0.997	0.994	0.992	0.990	0.995	0.997	0.989	0.994	0.995	0.990	0.992	0.957	0.994	0.990	0.989	0.992	0.989	0.991	0.990
Celebrity (E)	0.986	0.991	0.992	0.992	0.988	0.993	0.000	0.991	0.994	0.994	0.996	0.993	0.986	0.718	0.987	0.993	0.990	0.994	0.683	0.998	0.995	0.990	0.996	0.990	0.619	0.990	0.998	0.991	0.995	0.992	0.988	0.993	0.987	0.993	0.996	0.976	0.989	0.990	0.990	0.998	0.995	0.990
Europe (W)	0.962	0.988	0.984	0.988	0.983	0.993	0.991	0.000	0.994	0.996	0.998	0.993	0.952	0.995	0.991	0.983	0.963	0.995	0.995	0.996	0.988	0.996	0.995	0.997	0.994	0.996	0.993	0.993	0.997	0.992	0.995	0.994	0.992	0.994	0.996	0.993	0.987	0.988	0.995	0.995	0.994	0.988
Motor (Sp)	0.997	0.998	0.993	0.995	0.986	0.995	0.994	0.994	0.000	0.994	0.998	0.997	0.995	0.991	0.993	0.994	0.993	0.987	0.990	0.992	0.994	0.995	0.989	0.986	0.994	0.985	0.996	0.988	0.995	0.991	0.984	0.996	0.992	0.995	0.999	0.910	0.994	0.991	0.994	0.995	0.996	0.988
Software (T)	0.992	0.994	0.987	0.994	0.994	0.988	0.994	0.996	0.994	0.000	0.954	0.981	0.994	0.993	0.996	0.994	0.996	0.993	0.993	0.998	0.997	0.981	0.996	0.993	0.995	0.992	0.991	0.995	0.914	0.995	0.995	0.996	0.912	0.919	0.995	0.992	0.997	0.992	0.992	0.997	0.984	0.994
Apple (T)	0.994	0.989	0.991	0.995	0.995	0.994	0.996	0.998	0.998	0.954	0.000	0.978	0.996	0.993	0.997	0.994	0.995	0.996	0.993	0.999	0.996	0.979	0.997	0.994	0.996	0.995	0.992	0.997	0.968	0.994	0.993	0.994	0.977	0.925	0.994	0.983	0.996	0.993	0.992	0.998	0.987	0.993
Comm (T)	0.989	0.988	0.988	0.994	0.994	0.990	0.993	0.993	0.997	0.981	0.978	0.000	0.989	0.991	0.995	0.991	0.994	0.991	0.989	0.996	0.990	0.923	0.996	0.994	0.987	0.991	0.982	0.995	0.987	0.988	0.989	0.991	0.980	0.519	0.995	0.948	0.995	0.989	0.990	0.995	0.979	0.991
Fashion (E)	0.992	0.998	0.987	0.997	0.991	0.993	0.986	0.952	0.995	0.994	0.996	0.989	0.000	0.983	0.993	0.991	0.994	0.992	0.989	0.997	0.994	0.989	0.998	0.996	0.988	0.992	0.990	0.993	0.994	0.989	0.988	0.994	0.987	0.990	0.996	0.984	0.996	0.989	0.990	0.996	0.988	0.989
Movies (E)	0.991	0.988	0.989	0.996	0.990	0.990	0.718	0.995	0.991	0.993	0.993	0.991	0.983	0.000	0.991	0.996	0.990	0.996	0.840	0.998	0.995	0.991	0.997	0.992	0.721	0.990	0.997	0.991	0.994	0.991	0.992	0.996	0.989	0.995	0.998	0.881	0.993	0.995	0.984	0.998	0.996	0.991
Space (Sc)	0.991	0.984	0.942	0.991	0.969	0.981	0.987	0.991	0.993	0.996	0.997	0.995	0.993	0.991	0.000	0.991	0.921	0.983	0.995	0.998	0.992	0.996	0.997	0.994	0.996	0.995	0.993	0.997	0.997	0.992	0.996	0.996	0.995	0.978	0.998	0.964	0.996	0.993	0.995	0.994	0.973	0.961
Asia (W)	0.976	0.991	0.980	0.951	0.994	0.976	0.993	0.983	0.994	0.994	0.994	0.991	0.991	0.996	0.991	0.000	0.983	0.995	0.995	0.996	0.965	0.992	0.996	0.997	0.995	0.995	0.990	0.996	0.994	0.990	0.993	0.992	0.993	0.991	0.994	0.994	0.986	0.971	0.987	0.990	0.990	0.979
Crimes/Trials (US)	0.987	0.990	0.990	0.992	0.989	0.989	0.990	0.963	0.993	0.996	0.995	0.994	0.994	0.990	0.921	0.983	0.000	0.994	0.958	0.997	0.991	0.995	0.997	0.992	0.990	0.991	0.994	0.991	0.995	0.956	0.991	0.995	0.994	0.993	0.995	0.992	0.988	0.992	0.992	0.998	0.992	0.980
Football (Sp)	0.993	0.992	0.991	0.995	0.964	0.993	0.994	0.995	0.987	0.993	0.996	0.991	0.992	0.996	0.983	0.995	0.994	0.000	0.989	0.995	0.997	0.994	0.992	0.979	0.992	0.974	0.996	0.971	0.992	0.989	0.979	0.993	0.993	0.976	0.997	0.960	0.996	0.994	0.995	0.997	0.978	0.963
Music (E)	0.993	0.991	0.994	0.995	0.994	0.993	0.683	0.995	0.990	0.993	0.993	0.989	0.989	0.840	0.995	0.995	0.958	0.989	0.000	0.998	0.996	0.991	0.998	0.990	0.762	0.986	0.996	0.989	0.996	0.991	0.988	0.992	0.933	0.991	0.996	0.958	0.992	0.993	0.991	0.998	0.996	0.988
Tennis (S)	0.996	0.995	0.996	0.997	0.987	0.998	0.998	0.996	0.992	0.998	0.999	0.996	0.997	0.998	0.998	0.996	0.997	0.995	0.998	0.000	0.990	0.988	0.978	0.978	0.986	0.970	0.996	0.982	0.991	0.990	0.981	0.993	0.994	0.998	1.000	0.990	0.986	0.989	0.993	0.995	0.993	0.983
Australia (W)	0.978	0.984	0.969	0.978	0.997	0.951	0.995	0.988	0.994	0.997	0.996	0.990	0.994	0.995	0.992	0.965	0.991	0.997	0.996	0.990	0.000	0.988	0.996	0.996	0.992	0.995	0.992	0.997	0.995	0.982	0.990	0.992	0.985	0.990	0.995	0.990	0.982	0.959	0.987	0.984	0.989	0.974
Earnings (B)	0.992	0.984	0.987	0.995	0.988	0.993	0.990	0.996	0.995	0.981	0.979	0.923	0.989	0.991	0.996	0.992	0.995	0.994	0.991	0.988	0.988	0.000	0.997	0.996	0.989	0.991	0.959	0.996	0.978	0.989	0.990	0.987	0.978	0.959	0.994	0.978	0.995	0.985	0.987	0.995	0.915	0.991
Golf (Sp)	0.992	0.995	0.995	0.995	0.994	0.994	0.996	0.995	0.989	0.996	0.997	0.996	0.998	0.997	0.997	0.996	0.997	0.992	0.998	0.978	0.996	0.997	0.000	0.985	0.994	0.987	0.997	0.989	0.993	0.992	0.984	0.996	0.992	0.994	1.000	0.986	0.997	0.992	0.997	0.998	0.996	0.989
NCAA (Sp)	0.997	0.993	0.995	0.995	0.992	0.997	0.990	0.997	0.986	0.993	0.994	0.994	0.996	0.992	0.994	0.997	0.992	0.979	0.990	0.978	0.996	0.996	0.985	0.000	0.992	0.985	0.996	0.978	0.994	0.990	0.949	0.994	0.995	0.996	0.999	0.991	0.993	0.992	0.994	0.996	0.996	0.989
TV (E)	0.995	0.991	0.991	0.992	0.989	0.994	0.619	0.994	0.994	0.995	0.996	0.987	0.988	0.721	0.996	0.995	0.990	0.992	0.762	0.986	0.992	0.989	0.994	0.992	0.000	0.984	0.991	0.989	0.996	0.987	0.985	0.988	0.896	0.993	0.998	0.969	0.990	0.986	0.979	0.994	0.995	0.982
Baseball (Sp)	0.993	0.988	0.992	0.996	0.984	0.992	0.990	0.996	0.985	0.992	0.995	0.991	0.992	0.990	0.995	0.995	0.991	0.974	0.986	0.970	0.995	0.991	0.987	0.985	0.984	0.000	0.997	0.973	0.991	0.989	0.973	0.992	0.991	0.993	0.996	0.985	0.993	0.992	0.992	0.998	0.994	0.987
Economy (B)	0.990	0.985	0.981	0.991	0.995	0.990	0.998	0.993	0.996	0.991	0.992	0.982	0.990	0.997	0.993	0.990	0.994	0.996	0.996	0.996	0.992	0.959	0.997	0.996	0.991	0.997	0.000	0.993	0.991	0.985	0.989	0.987	0.988	0.987	0.995	0.985	0.992	0.942	0.989	0.986	0.888	0.983
Hockey (S)	0.995	0.990	0.995	0.997	0.981	0.995	0.991	0.993	0.988	0.995	0.997	0.995	0.993	0.991	0.997	0.996	0.991	0.971	0.989	0.982	0.997	0.996	0.989	0.978	0.989	0.973	0.993	0.000	0.994	0.988	0.955	0.994	0.988	0.990	1.000	0.989	0.993	0.993	0.993	0.999	0.989	0.987
Open source (T)	0.997	0.989	0.992	0.994	0.993	0.997	0.995	0.997	0.995	0.914	0.968	0.987	0.994	0.994	0.997	0.994	0.995	0.992	0.996	0.991	0.995	0.978	0.993	0.994	0.996	0.991	0.991	0.994	0.000	0.995	0.997	0.996	0.983	0.989	0.996	0.992	0.997	0.995	0.993	0.997	0.995	0.991
Weather (Sc)	0.952	0.984	0.977	0.994	0.995	0.989	0.992	0.992	0.991	0.995	0.994	0.988	0.989	0.991	0.992	0.990	0.956	0.989	0.991	0.990	0.982	0.989	0.992	0.990	0.987	0.989	0.985	0.988	0.995	0.000	0.987	0.969	0.991	0.992	0.996	0.986	0.985	0.905	0.990	0.989	0.993	0.983
Basketball (Sp)	0.993	0.988	0.988	0.997	0.985	0.994	0.988	0.995	0.984	0.995	0.993	0.989	0.988	0.992	0.996	0.993	0.991	0.979	0.988	0.981	0.990	0.990	0.984	0.949	0.985	0.973	0.989	0.955	0.997	0.987	0.000	0.991	0.986	0.994	0.998	0.985	0.994	0.990	0.993	0.997	0.993	0.990
Education (US)	0.994	0.995	0.989	0.995	0.995	0.995	0.993	0.994	0.996	0.996	0.994	0.991	0.994	0.996	0.996	0.992	0.995	0.993	0.992	0.993	0.992	0.987	0.996	0.994	0.988	0.992	0.987	0.994	0.996	0.969	0.991	0.000	0.989	0.991	0.994	0.989	0.994	0.925	0.985	0.992	0.991	0.973
Internet (T)	0.992	0.981	0.981	0.994	0.993	0.990	0.987	0.992	0.992	0.912	0.977	0.980	0.987	0.989	0.995	0.993	0.994	0.993	0.933	0.994	0.985	0.978	0.992	0.995	0.896	0.991	0.988	0.988	0.983	0.991	0.986	0.989	0.000	0.894	0.997	0.893	0.995	0.993	0.992	0.996	0.987	0.990
Personal (T)	0.991	0.991	0.988	0.993	0.964	0.992	0.993	0.994	0.995	0.919	0.925	0.519	0.990	0.995	0.978	0.991	0.993	0.976	0.991	0.998	0.990	0.959	0.994	0.996	0.993	0.993	0.987	0.990	0.989	0.992	0.994	0.991	0.894	0.000	0.996	0.934	0.993	0.989	0.990	0.993	0.944	0.963
Biotech (T)	0.995	0.992	0.992	0.994	0.994	0.957	0.996	0.996	0.999	0.995	0.994	0.995	0.996	0.998	0.998	0.994	0.995	0.997	0.996	1.000	0.995	0.994	1.000	0.999	0.998	0.996	0.995	1.000	0.996	0.996	0.998	0.994	0.997	0.996	0.000	0.995	0.992	0.982	0.995	0.991	0.992	0.979
Industry (E)	0.994	0.988	0.984	0.994	0.946	0.994	0.976	0.993	0.910	0.992	0.983	0.948	0.984	0.881	0.964	0.994	0.992	0.960	0.958	0.990	0.990	0.978	0.986	0.991	0.969	0.985	0.985	0.989	0.992	0.986	0.985	0.989	0.893	0.934	0.995	0.000	0.996	0.996	0.995	0.993	0.998	0.979	0.965
Latin (W)	0.985	0.992	0.970	0.978	0.993	0.990	0.989	0.987	0.994	0.997	0.996	0.995	0.996	0.993	0.996	0.986	0.988	0.996	0.992	0.986	0.982	0.995	0.997	0.993	0.990	0.993	0.992	0.993	0.997	0.985	0.994	0.994	0.995	0.993	0.992	0.996	0.000	0.984	0.963	0.992	0.996	0.981
Politics (US)	0.949	0.981	0.961	0.977	0.994	0.989	0.990	0.988	0.991	0.992	0.993	0.989	0.989	0.995	0.993	0.971	0.992	0.994	0.993	0.989	0.959	0.985	0.992	0.992	0.986	0.992	0.942	0.993	0.995	0.905	0.990	0.925	0.993	0.989	0.982	0.995	0.984	0.000	0.979	0.977	0.987	0.948
Books (E)	0.993	0.993	0.988	0.992	0.994	0.992	0.990	0.995	0.994	0.992	0.992	0.990	0.990	0.984	0.995	0.987	0.992	0.995	0.991	0.993	0.987	0.987	0.997	0.994	0.979	0.992	0.989	0.993	0.993	0.990	0.993	0.985	0.992	0.990	0.995	0.993	0.963	0.979	0.000	0.993	0.994	0.984
Energy (Sc)	0.989	0.996	0.957	0.989	0.998	0.989	0.998	0.995	0.995	0.997	0.998	0.995	0.996	0.998	0.994	0.990	0.998	0.997	0.998	0.995	0.984	0.995	0.998	0.996	0.994	0.998	0.986	0.999	0.997	0.989	0.997	0.992	0.996	0.993	0.991	0.998	0.992	0.977	0.993	0.000	0.979	0.975
Markets (B)	0.989	0.980	0.984	0.997	0.972	0.991	0.995	0.994	0.996	0.984	0.987	0.979	0.988	0.996	0.973	0.990	0.992	0.978	0.996	0.993	0.989	0.915	0.996	0.996	0.995	0.994	0.888	0.989	0.995	0.993	0.993	0.991	0.987	0.944	0.992	0.979	0.996	0.987	0.994	0.979	0.000	0.966
Religion (US)	0.982	0.976	0.971	0.983	0.952	0.990	0.990	0.988	0.988	0.994	0.993	0.991	0.989	0.991	0.961	0.979	0.980	0.963	0.988	0.983	0.974	0.991	0.989	0.989	0.982	0.987	0.983	0.987	0.991	0.983	0.990	0.973	0.990	0.963	0.979	0.965	0.981	0.948	0.984	0.975	0.966	0.000

Table B.3: SID Distance between Yahoo! News Graphs

	Africa (W)	Canada (W)	Environment (Sc)	Mideast (W)	Soccer (Sp)	Animals (Sc)	Celebrity (E)	Europe (W)	Motor (Sp)	Software (T)	Apple (T)	Comm (T)	Fashion (E)	Movies (E)	Space (Sc)	Asia (W)	Crimes/Trials (US)	Football (Sp)	Music (E)	Tennis (S)	Australia (W)	Earnings (B)	Golf (Sp)	NCAA (Sp)	TV (E)	Baseball (Sp)	Economy (B)	Hockey (S)	Open source (T)	Weather (Sc)	Basketball (Sp)	Education (US)	Internet (T)	Personal (T)	Biotech (T)	Industry (E)	Latin (W)	Politics (US)	Books (E)	Energy (Sc)	Markets (B)	Religion (US)
Africa (W)	0.000	0.988	0.960	0.951	0.992	0.971	0.977	0.947	0.987	0.985	0.990	0.978	0.987	0.985	0.979	0.952	0.981	0.982	0.986	0.992	0.957	0.983	0.982	0.986	0.986	0.981	0.977	0.986	0.985	0.935	0.986	0.986	0.983	0.985	0.991	0.986	0.960	0.934	0.985	0.977	0.983	0.970
Canada (W)	0.988	0.000	0.988	0.993	0.988	0.992	0.993	0.989	0.994	0.994	0.990	0.989	0.995	0.994	0.991	0.994	0.990	0.992	0.994	0.995	0.985	0.992	0.993	0.990	0.992	0.992	0.981	0.993	0.992	0.985	0.993	0.992	0.989	0.991	0.985	0.993	0.996	0.988	0.993	0.987	0.986	0.991
Environment (Sc)	0.960	0.988	0.000	0.975	0.993	0.949	0.982	0.981	0.979	0.973	0.980	0.973	0.979	0.980	0.931	0.959	0.982	0.983	0.986	0.988	0.950	0.972	0.991	0.975	0.979	0.978	0.958	0.987	0.981	0.953	0.982	0.977	0.970	0.976	0.985	0.975	0.960	0.943	0.977	0.950	0.962	0.970
Mideast (W)	0.951	0.993	0.975	0.000	0.991	0.983	0.986	0.967	0.981	0.988	0.983	0.982	0.988	0.987	0.981	0.918	0.979	0.984	0.986	0.992	0.953	0.986	0.988	0.986	0.988	0.987	0.982	0.992	0.989	0.983	0.993	0.987	0.990	0.981	0.992	0.990	0.963	0.958	0.988	0.979	0.985	0.976
Soccer (Sp)	0.992	0.988	0.993	0.991	0.000	0.994	0.992	0.987	0.988	0.991	0.991	0.990	0.993	0.995	0.980	0.992	0.985	0.968	0.994	0.974	0.991	0.989	0.988	0.982	0.991	0.985	0.991	0.969	0.995	0.991	0.982	0.991	0.992	0.979	0.985	0.982	0.988	0.993	0.994	0.990	0.981	0.981
Animals (Sc)	0.971	0.992	0.949	0.983	0.994	0.000	0.984	0.985	0.989	0.980	0.988	0.978	0.989	0.988	0.969	0.947	0.975	0.984	0.989	0.993	0.944	0.985	0.990	0.987	0.991	0.981	0.978	0.986	0.986	0.966	0.984	0.984	0.984	0.980	0.960	0.990	0.975	0.977	0.982	0.984	0.982	0.980
Celebrity (E)	0.977	0.993	0.982	0.986	0.992	0.984	0.000	0.981	0.983	0.988	0.992	0.982	0.982	0.804	0.983	0.982	0.975	0.981	0.745	0.994	0.980	0.982	0.992	0.976	0.719	0.978	0.990	0.983	0.990	0.979	0.979	0.989	0.976	0.987	0.990	0.967	0.980	0.977	0.974	0.992	0.990	0.976
Europe (W)	0.947	0.989	0.981	0.967	0.987	0.985	0.981	0.000	0.984	0.986	0.994	0.985	0.959	0.993	0.985	0.962	0.951	0.986	0.989	0.988	0.959	0.993	0.992	0.986	0.991	0.985	0.982	0.983	0.990	0.984	0.983	0.992	0.988	0.987	0.991	0.992	0.965	0.977	0.986	0.984	0.989	0.982
Motor (Sp)	0.987	0.994	0.979	0.981	0.988	0.989	0.983	0.984	0.000	0.982	0.985	0.985	0.985	0.983	0.987	0.989	0.983	0.963	0.980	0.975	0.983	0.989	0.969	0.959	0.984	0.958	0.983	0.958	0.984	0.983	0.957	0.993	0.978	0.978	0.996	0.942	0.980	0.981	0.987	0.988	0.986	0.983
Software (T)	0.985	0.994	0.973	0.988	0.991	0.980	0.988	0.986	0.982	0.000	0.907	0.947	0.983	0.988	0.985	0.982	0.984	0.982	0.989	0.989	0.979	0.954	0.993	0.985	0.983	0.975	0.970	0.986	0.872	0.981	0.987	0.989	0.886	0.905	0.988	0.981	0.979	0.971	0.981	0.984	0.966	0.988
Apple (T)	0.990	0.990	0.980	0.983	0.991	0.988	0.992	0.994	0.985	0.907	0.000	0.941	0.989	0.986	0.987	0.990	0.988	0.986	0.987	0.994	0.985	0.941	0.992	0.986	0.984	0.983	0.965	0.990	0.932	0.986	0.986	0.984	0.960	0.897	0.988	0.977	0.988	0.984	0.986	0.987	0.965	0.985
Comm (T)	0.978	0.989	0.973	0.982	0.990	0.978	0.982	0.985	0.985	0.947	0.941	0.000	0.986	0.987	0.976	0.978	0.983	0.983	0.982	0.992	0.970	0.888	0.989	0.984	0.979	0.981	0.949	0.986	0.961	0.970	0.986	0.984	0.960	0.664	0.985	0.955	0.983	0.977	0.983	0.980	0.944	0.980
Fashion (E)	0.987	0.995	0.979	0.988	0.993	0.989	0.982	0.959	0.985	0.983	0.989	0.986	0.000	0.976	0.989	0.986	0.989	0.986	0.980	0.991	0.985	0.986	0.994	0.985	0.978	0.983	0.989	0.985	0.986	0.981	0.985	0.992	0.975	0.989	0.994	0.977	0.989	0.979	0.984	0.989	0.985	0.985
Movies (E)	0.985	0.994	0.980	0.987	0.995	0.988	0.804	0.993	0.983	0.988	0.986	0.987	0.976	0.000	0.986	0.983	0.982	0.986	0.866	0.992	0.984	0.986	0.994	0.977	0.788	0.977	0.993	0.984	0.988	0.980	0.983	0.986	0.980	0.987	0.992	0.891	0.984	0.979	0.977	0.993	0.989	0.975
Space (Sc)	0.979	0.991	0.931	0.981	0.980	0.969	0.983	0.985	0.987	0.985	0.987	0.976	0.989	0.986	0.000	0.976	0.914	0.966	0.986	0.987	0.973	0.987	0.993	0.978	0.989	0.983	0.982	0.985	0.985	0.969	0.983	0.989	0.983	0.953	0.989	0.957	0.979	0.972	0.990	0.974	0.942	0.947
Asia (W)	0.952	0.994	0.959	0.918	0.992	0.947	0.982	0.962	0.989	0.982	0.990	0.978	0.986	0.983	0.976	0.000	0.959	0.988	0.989	0.989	0.929	0.977	0.991	0.982	0.987	0.985	0.972	0.989	0.986	0.966	0.987	0.983	0.985	0.982	0.992	0.988	0.961	0.946	0.984	0.973	0.973	0.970
Crimes/Trials (US)	0.981	0.990	0.982	0.979	0.985	0.975	0.975	0.951	0.983	0.984	0.988	0.983	0.989	0.982	0.914	0.959	0.000	0.979	0.968	0.989	0.971	0.987	0.988	0.979	0.985	0.982	0.987	0.988	0.991	0.955	0.982	0.982	0.985	0.984	0.987	0.987	0.973	0.978	0.983	0.989	0.988	0.979
Football (Sp)	0.982	0.992	0.983	0.984	0.968	0.984	0.981	0.986	0.963	0.982	0.986	0.983	0.986	0.986	0.966	0.988	0.979	0.000	0.983	0.987	0.984	0.985	0.979	0.950	0.984	0.939	0.980	0.917	0.987	0.975	0.941	0.988	0.982	0.966	0.993	0.962	0.987	0.987	0.985	0.988	0.971	0.964
Music (E)	0.986	0.994	0.986	0.986	0.994	0.989	0.745	0.989	0.980	0.989	0.987	0.982	0.980	0.866	0.986	0.989	0.968	0.983	0.000	0.990	0.990	0.989	0.990	0.974	0.810	0.976	0.991	0.984	0.988	0.978	0.983	0.990	0.941	0.988	0.990	0.958	0.985	0.982	0.982	0.994	0.992	0.979
Tennis (S)	0.992	0.995	0.988	0.992	0.974	0.993	0.994	0.988	0.975	0.989	0.994	0.992	0.991	0.992	0.987	0.989	0.989	0.987	0.990	0.000	0.989	0.986	0.963	0.970	0.991	0.975	0.991	0.973	0.991	0.990	0.969	0.991	0.992	0.993	0.998	0.990	0.985	0.991	0.991	0.987	0.992	0.989
Australia (W)	0.957	0.985	0.950	0.953	0.991	0.944	0.980	0.959	0.983	0.979	0.985	0.970	0.985	0.984	0.973	0.929	0.971	0.984	0.990	0.989	0.000	0.970	0.987	0.974	0.983	0.985	0.969	0.985	0.982	0.960	0.983	0.982	0.975	0.972	0.990	0.986	0.957	0.945	0.974	0.966	0.974	0.976
Earnings (B)	0.983	0.992	0.972	0.986	0.989	0.985	0.982	0.993	0.989	0.954	0.941	0.888	0.986	0.986	0.987	0.977	0.987	0.985	0.989	0.986	0.970	0.000	0.993	0.984	0.981	0.982	0.920	0.985	0.960	0.976	0.981	0.979	0.965	0.928	0.992	0.970	0.981	0.978	0.984	0.981	0.892	0.986
Golf (Sp)	0.982	0.993	0.991	0.988	0.988	0.990	0.992	0.992	0.969	0.993	0.992	0.989	0.994	0.994	0.993	0.991	0.988	0.979	0.990	0.963	0.987	0.993	0.000	0.971	0.993	0.977	0.993	0.973	0.992	0.988	0.975	0.991	0.988	0.990	0.998	0.992	0.991	0.993	0.994	0.991	0.993	0.990
NCAA (Sp)	0.986	0.990	0.975	0.986	0.982	0.987	0.976	0.986	0.959	0.985	0.986	0.984	0.985	0.977	0.978	0.982	0.979	0.950	0.974	0.970	0.974	0.984	0.971	0.000	0.977	0.943	0.985	0.946	0.983	0.966	0.900	0.983	0.978	0.983	0.993	0.979	0.981	0.971	0.983	0.987	0.988	0.975
TV (E)	0.986	0.992	0.979	0.988	0.991	0.991	0.719	0.991	0.984	0.983	0.984	0.979	0.978	0.788	0.989	0.987	0.985	0.984	0.810	0.991	0.983	0.981	0.993	0.977	0.000	0.976	0.988	0.981	0.988	0.982	0.977	0.986	0.889	0.976	0.993	0.953	0.987	0.974	0.981	0.995	0.986	0.982
Baseball (Sp)	0.981	0.992	0.978	0.987	0.985	0.981	0.978	0.985	0.958	0.975	0.983	0.981	0.983	0.977	0.983	0.985	0.982	0.939	0.976	0.975	0.985	0.982	0.977	0.943	0.976	0.000	0.984	0.941	0.981	0.974	0.928	0.985	0.980	0.983	0.995	0.976	0.987	0.975	0.980	0.989	0.982	0.979
Economy (B)	0.977	0.981	0.958	0.982	0.991	0.978	0.990	0.982	0.983	0.970	0.965	0.949	0.989	0.993	0.982	0.972	0.987	0.980	0.991	0.991	0.969	0.920	0.993	0.985	0.988	0.984	0.000	0.986	0.980	0.975	0.980	0.979	0.981	0.961	0.991	0.980	0.979	0.942	0.985	0.970	0.851	0.980
Hockey (S)	0.986	0.993	0.987	0.992	0.969	0.986	0.983	0.983	0.958	0.986	0.990	0.986	0.985	0.984	0.985	0.989	0.988	0.917	0.984	0.973	0.985	0.985	0.973	0.946	0.981	0.941	0.986	0.000	0.986	0.985	0.909	0.993	0.982	0.985	0.996	0.984	0.988	0.988	0.986	0.992	0.987	0.984
Open source (T)	0.985	0.992	0.981	0.989	0.995	0.986	0.990	0.990	0.984	0.872	0.932	0.961	0.986	0.988	0.985	0.986	0.991	0.987	0.988	0.991	0.982	0.960	0.992	0.983	0.988	0.981	0.980	0.986	0.000	0.980	0.985	0.989	0.964	0.959	0.993	0.982	0.986	0.981	0.983	0.985	0.973	0.982
Weather (Sc)	0.935	0.985	0.953	0.983	0.991	0.966	0.979	0.984	0.983	0.981	0.986	0.970	0.981	0.980	0.969	0.966	0.955	0.975	0.978	0.990	0.960	0.976	0.988	0.966	0.982	0.974	0.975	0.985	0.980	0.000	0.970	0.954	0.978	0.977	0.992	0.984	0.963	0.895	0.982	0.978	0.977	0.975
Basketball (Sp)	0.986	0.993	0.982	0.993	0.982	0.984	0.979	0.983	0.957	0.987	0.986	0.986	0.985	0.983	0.983	0.987	0.982	0.941	0.983	0.969	0.983	0.981	0.975	0.900	0.977	0.928	0.980	0.909	0.985	0.970	0.000	0.989	0.979	0.985	0.995	0.982	0.987	0.978	0.984	0.992	0.983	0.984
Education (US)	0.986	0.992	0.977	0.987	0.991	0.984	0.989	0.992	0.993	0.989	0.984	0.984	0.992	0.986	0.989	0.983	0.982	0.988	0.990	0.991	0.982	0.979	0.991	0.983	0.986	0.985	0.979	0.993	0.989	0.954	0.989	0.000	0.987	0.987	0.982	0.990	0.985	0.947	0.982	0.986	0.989	0.980
Internet (T)	0.983	0.989	0.970	0.990	0.992	0.984	0.976	0.988	0.978	0.886	0.960	0.960	0.975	0.980	0.983	0.985	0.985	0.982	0.941	0.992	0.975	0.965	0.988	0.978	0.889	0.980	0.981	0.982	0.964	0.978	0.979	0.987	0.000	0.895	0.994	0.912	0.989	0.971	0.980	0.988	0.973	0.984
Personal (T)	0.985	0.991	0.976	0.981	0.979	0.980	0.987	0.987	0.978	0.905	0.897	0.664	0.989	0.987	0.953	0.982	0.984	0.966	0.988	0.993	0.972	0.928	0.990	0.983	0.976	0.983	0.961	0.985	0.959	0.977	0.985	0.987	0.895	0.000	0.989	0.941	0.983	0.977	0.982	0.979	0.921	0.959
Biotech (T)	0.991	0.985	0.985	0.992	0.985	0.960	0.990	0.991	0.996	0.988	0.988	0.985	0.994	0.992	0.989	0.992	0.987	0.993	0.990	0.998	0.990	0.992	0.998	0.993	0.993	0.995	0.991	0.996	0.993	0.992	0.995	0.982	0.994	0.989	0.000	0.994	0.990	0.985	0.990	0.976	0.995	0.985
Industry (E)	0.986	0.993	0.975	0.990	0.982	0.990	0.967	0.992	0.942	0.981	0.977	0.955	0.977	0.891	0.957	0.988	0.987	0.962	0.958	0.990	0.986	0.970	0.992	0.979	0.953	0.976	0.980	0.984	0.982	0.984	0.982	0.990	0.912	0.941	0.994	0.000	0.989	0.984	0.984	0.987	0.949	0.927
Latin (W)	0.960	0.996	0.960	0.963	0.988	0.975	0.980	0.965	0.980	0.979	0.988	0.983	0.989	0.984	0.979	0.961	0.973	0.987	0.985	0.985	0.957	0.981	0.991	0.981	0.987	0.987	0.979	0.988	0.986	0.963	0.987	0.985	0.989	0.983	0.990	0.989	0.000	0.962	0.959	0.978	0.985	0.975
Politics (US)	0.934	0.988	0.943	0.958	0.993	0.977	0.977	0.977	0.981	0.971	0.984	0.977	0.979	0.979	0.972	0.946	0.978	0.987	0.982	0.991	0.945	0.978	0.993	0.971	0.974	0.975	0.942	0.988	0.981	0.895	0.978	0.947	0.971	0.977	0.985	0.984	0.962	0.000	0.975	0.970	0.968	0.940
Books (E)	0.985	0.993	0.977	0.988	0.994	0.982	0.974	0.986	0.987	0.981	0.986	0.983	0.984	0.977	0.990	0.984	0.983	0.985	0.982	0.991	0.974	0.984	0.994	0.983	0.981	0.980	0.985	0.986	0.983	0.982	0.984	0.982	0.980	0.982	0.990	0.984	0.959	0.975	0.000	0.989	0.987	0.979
Energy (Sc)	0.977	0.987	0.950	0.979	0.990	0.984	0.992	0.984	0.988	0.984	0.987	0.980	0.989	0.993	0.974	0.973	0.989	0.988	0.994	0.987	0.966	0.981	0.991	0.987	0.995	0.989	0.970	0.992	0.985	0.978	0.992	0.986	0.988	0.979	0.976	0.987	0.978	0.970	0.989	0.000	0.985	0.982
Markets (B)	0.983	0.986	0.962	0.985	0.981	0.982	0.990	0.989	0.986	0.966	0.965	0.944	0.985	0.989	0.942	0.973	0.988	0.971	0.992	0.992	0.974	0.892	0.993	0.988	0.986	0.982	0.851	0.987	0.973	0.977	0.983	0.989	0.973	0.921	0.995	0.949	0.985	0.968	0.987	0.985	0.000	0.953
Religion (US)	0.970	0.991	0.970	0.976	0.981	0.980	0.976	0.982	0.983	0.988	0.985	0.980	0.985	0.975	0.947	0.970	0.979	0.964	0.979	0.989	0.976	0.986	0.990	0.975	0.982	0.979	0.980	0.984	0.982	0.975	0.984	0.980	0.984	0.959	0.985	0.927	0.975	0.940	0.979	0.982	0.953	0.000

Table B.4: mcs Distance between Node-Edge Samples (50%)

	Africa (W)	Canada (W)	Environment (Sc)	Mideast (W)	Soccer (Sp)	Animals (Sc)	Celebrity (E)	Europe (W)	Motor (Sp)	Software (T)	Apple (T)	Comm (T)	Fashion (E)	Movies (E)	Space (Sc)	Asia (W)	Crimes/Trials (US)	Football (Sp)	Music (E)	Tennis (S)	Australia (W)	Earnings (B)	Golf (Sp)	NCAA (Sp)	TV (E)	Baseball (Sp)	Economy (B)	Hockey (S)	Open source (T)	Weather (Sc)	Basketball (Sp)	Education (US)	Internet (T)	Personal (T)	Biotech (T)	Industry (E)	Latin (W)	Politics (US)	Books (E)	Energy (Sc)	Markets (B)	Religion (US)
Africa (W)	0.000	0.990	0.977	0.968	0.994	0.984	0.986	0.970	0.993	0.992	0.994	0.988	0.993	0.991	0.988	0.972	0.988	0.989	0.992	0.994	0.974	0.990	0.988	0.992	0.992	0.990	0.986	0.992	0.992	0.963	0.993	0.991	0.990	0.991	0.993	0.991	0.976	0.960	0.992	0.983	0.991	0.980
Canada (W)	0.990	0.000	0.991	0.995	0.993	0.994	0.994	0.992	0.995	0.995	0.993	0.991	0.996	0.995	0.993	0.995	0.993	0.994	0.995	0.997	0.989	0.994	0.995	0.992	0.994	0.993	0.986	0.995	0.994	0.988	0.995	0.995	0.991	0.993	0.991	0.994	0.997	0.990	0.994	0.992	0.989	0.992
Environment (Sc)	0.977	0.991	0.000	0.983	0.994	0.970	0.989	0.989	0.988	0.985	0.988	0.984	0.988	0.987	0.960	0.976	0.988	0.989	0.992	0.991	0.968	0.983	0.994	0.987	0.988	0.988	0.973	0.992	0.989	0.972	0.990	0.985	0.983	0.986	0.988	0.983	0.976	0.967	0.988	0.962	0.978	0.980
Mideast (W)	0.968	0.995	0.983	0.000	0.994	0.990	0.990	0.979	0.989	0.993	0.990	0.989	0.992	0.991	0.988	0.948	0.989	0.991	0.991	0.995	0.972	0.992	0.993	0.991	0.992	0.991	0.990	0.995	0.993	0.989	0.995	0.993	0.994	0.989	0.995	0.993	0.977	0.972	0.993	0.986	0.991	0.982
Soccer (Sp)	0.994	0.993	0.994	0.994	0.000	0.995	0.993	0.990	0.991	0.993	0.993	0.992	0.994	0.996	0.984	0.994	0.989	0.977	0.995	0.985	0.993	0.992	0.992	0.986	0.993	0.988	0.993	0.977	0.996	0.993	0.986	0.994	0.993	0.984	0.991	0.984	0.991	0.994	0.995	0.994	0.985	0.983
Animals (Sc)	0.984	0.994	0.970	0.990	0.995	0.000	0.990	0.992	0.994	0.989	0.993	0.988	0.994	0.992	0.983	0.970	0.984	0.990	0.993	0.995	0.966	0.992	0.994	0.993	0.994	0.990	0.987	0.992	0.992	0.981	0.992	0.990	0.991	0.988	0.970	0.993	0.986	0.986	0.990	0.989	0.990	0.986
Celebrity (E)	0.986	0.994	0.989	0.990	0.993	0.990	0.000	0.988	0.989	0.993	0.995	0.988	0.990	0.878	0.989	0.989	0.982	0.987	0.844	0.995	0.986	0.988	0.994	0.985	0.822	0.986	0.993	0.989	0.994	0.987	0.987	0.992	0.986	0.991	0.992	0.979	0.987	0.986	0.984	0.994	0.994	0.986
Europe (W)	0.970	0.992	0.989	0.979	0.990	0.992	0.988	0.000	0.991	0.992	0.997	0.992	0.975	0.995	0.992	0.979	0.969	0.991	0.993	0.992	0.975	0.996	0.995	0.992	0.994	0.991	0.989	0.990	0.994	0.991	0.991	0.995	0.993	0.993	0.993	0.994	0.980	0.986	0.992	0.989	0.994	0.988
Motor (Sp)	0.993	0.995	0.988	0.989	0.991	0.994	0.989	0.991	0.000	0.989	0.992	0.992	0.991	0.988	0.992	0.994	0.990	0.977	0.987	0.982	0.991	0.994	0.979	0.975	0.990	0.975	0.990	0.975	0.991	0.990	0.975	0.996	0.987	0.988	0.997	0.957	0.989	0.989	0.992	0.991	0.992	0.988
Software (T)	0.992	0.995	0.985	0.993	0.993	0.989	0.993	0.992	0.989	0.000	0.941	0.969	0.990	0.992	0.992	0.990	0.990	0.989	0.993	0.992	0.988	0.973	0.995	0.992	0.990	0.987	0.982	0.992	0.924	0.990	0.993	0.993	0.931	0.942	0.991	0.987	0.988	0.983	0.989	0.988	0.981	0.992
Apple (T)	0.994	0.993	0.988	0.990	0.993	0.993	0.995	0.997	0.992	0.941	0.000	0.964	0.993	0.991	0.993	0.994	0.993	0.992	0.991	0.996	0.992	0.966	0.995	0.992	0.990	0.990	0.980	0.995	0.958	0.992	0.992	0.991	0.974	0.939	0.992	0.983	0.993	0.990	0.992	0.991	0.979	0.989
Comm (T)	0.988	0.991	0.984	0.989	0.992	0.988	0.988	0.992	0.992	0.969	0.964	0.000	0.992	0.991	0.987	0.988	0.990	0.990	0.989	0.994	0.982	0.933	0.993	0.991	0.987	0.989	0.969	0.992	0.978	0.984	0.992	0.990	0.975	0.780	0.989	0.968	0.991	0.986	0.991	0.985	0.968	0.986
Fashion (E)	0.993	0.996	0.988	0.992	0.994	0.994	0.990	0.975	0.991	0.990	0.993	0.992	0.000	0.985	0.994	0.992	0.992	0.991	0.988	0.993	0.991	0.991	0.996	0.992	0.988	0.990	0.993	0.990	0.992	0.989	0.991	0.995	0.986	0.993	0.995	0.985	0.993	0.988	0.991	0.992	0.991	0.991
Movies (E)	0.991	0.995	0.987	0.991	0.996	0.992	0.878	0.995	0.988	0.992	0.991	0.991	0.985	0.000	0.991	0.989	0.987	0.990	0.918	0.993	0.989	0.991	0.996	0.986	0.863	0.985	0.995	0.989	0.993	0.987	0.989	0.990	0.987	0.991	0.994	0.933	0.989	0.987	0.985	0.994	0.993	0.986
Space (Sc)	0.988	0.993	0.960	0.988	0.984	0.983	0.989	0.992	0.992	0.992	0.993	0.987	0.994	0.991	0.000	0.986	0.945	0.979	0.991	0.990	0.984	0.992	0.995	0.987	0.993	0.990	0.989	0.991	0.992	0.983	0.991	0.993	0.990	0.973	0.992	0.970	0.988	0.983	0.994	0.982	0.968	0.964
Asia (W)	0.972	0.995	0.976	0.948	0.994	0.970	0.989	0.979	0.994	0.990	0.994	0.988	0.992	0.989	0.986	0.000	0.975	0.993	0.993	0.992	0.958	0.987	0.994	0.989	0.992	0.991	0.984	0.993	0.992	0.981	0.993	0.990	0.991	0.990	0.994	0.992	0.978	0.966	0.991	0.981	0.985	0.979
Crimes/Trials (US)	0.988	0.993	0.988	0.989	0.989	0.984	0.982	0.969	0.990	0.990	0.993	0.990	0.992	0.987	0.945	0.975	0.000	0.989	0.978	0.993	0.983	0.993	0.993	0.986	0.990	0.989	0.993	0.993	0.995	0.971	0.989	0.990	0.990	0.991	0.991	0.990	0.983	0.985	0.989	0.993	0.993	0.984
Football (Sp)	0.989	0.994	0.989	0.991	0.977	0.990	0.987	0.991	0.977	0.989	0.992	0.990	0.991	0.990	0.979	0.993	0.989	0.000	0.988	0.991	0.991	0.991	0.988	0.967	0.989	0.960	0.989	0.951	0.992	0.985	0.963	0.993	0.988	0.980	0.995	0.971	0.992	0.992	0.991	0.992	0.982	0.974
Music (E)	0.992	0.995	0.992	0.991	0.995	0.993	0.844	0.993	0.987	0.993	0.991	0.989	0.988	0.918	0.991	0.993	0.978	0.988	0.000	0.992	0.994	0.993	0.993	0.985	0.887	0.985	0.994	0.989	0.993	0.986	0.989	0.993	0.965	0.992	0.992	0.973	0.991	0.990	0.989	0.995	0.995	0.987
Tennis (S)	0.994	0.997	0.991	0.995	0.985	0.995	0.995	0.992	0.982	0.992	0.996	0.994	0.993	0.993	0.990	0.992	0.993	0.991	0.992	0.000	0.992	0.990	0.976	0.977	0.993	0.981	0.994	0.981	0.993	0.993	0.977	0.995	0.993	0.995	0.999	0.992	0.989	0.993	0.994	0.993	0.994	0.991
Australia (W)	0.974	0.989	0.968	0.972	0.993	0.966	0.986	0.975	0.991	0.988	0.992	0.982	0.991	0.989	0.984	0.958	0.983	0.991	0.994	0.992	0.000	0.983	0.992	0.984	0.989	0.991	0.983	0.992	0.989	0.976	0.990	0.989	0.984	0.984	0.993	0.990	0.975	0.964	0.984	0.976	0.985	0.982
Earnings (B)	0.990	0.994	0.983	0.992	0.992	0.992	0.988	0.996	0.994	0.973	0.966	0.933	0.991	0.991	0.992	0.987	0.993	0.991	0.993	0.990	0.983	0.000	0.995	0.990	0.988	0.989	0.953	0.992	0.977	0.986	0.989	0.987	0.978	0.960	0.994	0.978	0.990	0.986	0.991	0.987	0.934	0.990
Golf (Sp)	0.988	0.995	0.994	0.993	0.992	0.994	0.994	0.995	0.979	0.995	0.995	0.993	0.996	0.996	0.995	0.994	0.993	0.988	0.993	0.976	0.992	0.995	0.000	0.980	0.995	0.984	0.996	0.983	0.995	0.992	0.983	0.995	0.992	0.993	0.999	0.993	0.995	0.995	0.996	0.995	0.996	0.992
NCAA (Sp)	0.992	0.992	0.987	0.991	0.986	0.993	0.985	0.992	0.975	0.992	0.992	0.991	0.992	0.986	0.987	0.989	0.986	0.967	0.985	0.977	0.984	0.990	0.980	0.000	0.987	0.967	0.991	0.965	0.990	0.980	0.940	0.989	0.988	0.990	0.994	0.986	0.989	0.984	0.991	0.991	0.993	0.983
TV (E)	0.992	0.994	0.988	0.992	0.993	0.994	0.822	0.994	0.990	0.990	0.990	0.987	0.988	0.863	0.993	0.992	0.990	0.989	0.887	0.993	0.989	0.988	0.995	0.987	0.000	0.986	0.992	0.988	0.993	0.989	0.986	0.990	0.935	0.985	0.994	0.969	0.992	0.985	0.989	0.996	0.992	0.989
Baseball (Sp)	0.990	0.993	0.988	0.991	0.988	0.990	0.986	0.991	0.975	0.987	0.990	0.989	0.990	0.985	0.990	0.991	0.989	0.960	0.985	0.981	0.991	0.989	0.984	0.967	0.986	0.000	0.990	0.963	0.989	0.985	0.958	0.990	0.988	0.990	0.996	0.983	0.992	0.986	0.989	0.992	0.990	0.986
Economy (B)	0.986	0.986	0.973	0.990	0.993	0.987	0.993	0.989	0.990	0.982	0.980	0.969	0.993	0.995	0.989	0.984	0.993	0.989	0.994	0.994	0.983	0.953	0.996	0.991	0.992	0.990	0.000	0.993	0.988	0.985	0.988	0.988	0.987	0.977	0.994	0.985	0.988	0.962	0.991	0.980	0.902	0.985
Hockey (S)	0.992	0.995	0.992	0.995	0.977	0.992	0.989	0.990	0.975	0.992	0.995	0.992	0.990	0.989	0.991	0.993	0.993	0.951	0.989	0.981	0.992	0.992	0.983	0.965	0.988	0.963	0.993	0.000	0.991	0.991	0.943	0.996	0.989	0.991	0.998	0.988	0.993	0.992	0.992	0.995	0.992	0.989
Open source (T)	0.992	0.994	0.989	0.993	0.996	0.992	0.994	0.994	0.991	0.924	0.958	0.978	0.992	0.993	0.992	0.992	0.995	0.992	0.993	0.993	0.989	0.977	0.995	0.990	0.993	0.989	0.988	0.991	0.000	0.989	0.992	0.993	0.978	0.975	0.995	0.987	0.992	0.989	0.991	0.989	0.985	0.988
Weather (Sc)	0.963	0.988	0.972	0.989	0.993	0.981	0.987	0.991	0.990	0.990	0.992	0.984	0.989	0.987	0.983	0.981	0.971	0.985	0.986	0.993	0.976	0.986	0.992	0.980	0.989	0.985	0.985	0.991	0.989	0.000	0.984	0.971	0.987	0.987	0.994	0.989	0.978	0.936	0.990	0.984	0.987	0.983
Basketball (Sp)	0.993	0.995	0.990	0.995	0.986	0.992	0.987	0.991	0.975	0.993	0.992	0.992	0.991	0.989	0.991	0.993	0.989	0.963	0.989	0.977	0.990	0.989	0.983	0.940	0.986	0.958	0.988	0.943	0.992	0.984	0.000	0.993	0.988	0.991	0.996	0.988	0.993	0.987	0.991	0.994	0.991	0.989
Education (US)	0.991	0.995	0.985	0.993	0.994	0.990	0.992	0.995	0.996	0.993	0.991	0.990	0.995	0.990	0.993	0.990	0.990	0.993	0.993	0.995	0.989	0.987	0.995	0.989	0.990	0.990	0.988	0.996	0.993	0.971	0.993	0.000	0.991	0.992	0.988	0.992	0.991	0.963	0.988	0.991	0.993	0.985
Internet (T)	0.990	0.991	0.983	0.994	0.993	0.991	0.986	0.993	0.987	0.931	0.974	0.975	0.986	0.987	0.990	0.991	0.990	0.988	0.965	0.993	0.984	0.978	0.992	0.988	0.935	0.988	0.987	0.989	0.978	0.987	0.988	0.991	0.000	0.932	0.995	0.940	0.993	0.984	0.989	0.991	0.984	0.990
Personal (T)	0.991	0.993	0.986	0.989	0.984	0.988	0.991	0.993	0.988	0.942	0.939	0.780	0.993	0.991	0.973	0.990	0.991	0.980	0.992	0.995	0.984	0.960	0.993	0.990	0.985	0.990	0.977	0.991	0.975	0.987	0.991	0.992	0.932	0.000	0.992	0.957	0.991	0.986	0.989	0.985	0.953	0.971
Biotech (T)	0.993	0.991	0.988	0.995	0.991	0.970	0.992	0.993	0.997	0.991	0.992	0.989	0.995	0.994	0.992	0.994	0.991	0.995	0.992	0.999	0.993	0.994	0.999	0.994	0.994	0.996	0.994	0.998	0.995	0.994	0.996	0.988	0.995	0.992	0.000	0.995	0.993	0.988	0.992	0.986	0.996	0.987
Industry (E)	0.991	0.994	0.983	0.993	0.984	0.993	0.979	0.994	0.957	0.987	0.983	0.968	0.985	0.933	0.970	0.992	0.990	0.971	0.973	0.992	0.990	0.978	0.993	0.986	0.969	0.983	0.985	0.988	0.987	0.989	0.988	0.992	0.940	0.957	0.995	0.000	0.992	0.990	0.989	0.989	0.965	0.958
Latin (W)	0.976	0.997	0.976	0.977	0.991	0.986	0.987	0.980	0.989	0.988	0.993	0.991	0.993	0.989	0.988	0.978	0.983	0.992	0.991	0.989	0.975	0.990	0.995	0.989	0.992	0.992	0.988	0.993	0.992	0.978	0.993	0.991	0.993	0.991	0.993	0.992	0.000	0.976	0.976	0.985	0.991	0.982
Politics (US)	0.960	0.990	0.967	0.972	0.994	0.986	0.986	0.986	0.989	0.983	0.990	0.986	0.988	0.987	0.983	0.966	0.985	0.992	0.990	0.993	0.964	0.986	0.995	0.984	0.985	0.986	0.962	0.992	0.989	0.936	0.987	0.963	0.984	0.986	0.988	0.990	0.976	0.000	0.986	0.977	0.981	0.960
Books (E)	0.992	0.994	0.988	0.993	0.995	0.990	0.984	0.992	0.992	0.989	0.992	0.991	0.991	0.985	0.994	0.991	0.989	0.991	0.989	0.994	0.984	0.991	0.996	0.991	0.989	0.989	0.991	0.992	0.991	0.990	0.991	0.988	0.989	0.989	0.992	0.989	0.976	0.986	0.000	0.992	0.993	0.986
Energy (Sc)	0.983	0.992	0.962	0.986	0.994	0.989	0.994	0.989	0.991	0.988	0.991	0.985	0.992	0.994	0.982	0.981	0.993	0.992	0.995	0.993	0.976	0.987	0.995	0.991	0.996	0.992	0.980	0.995	0.989	0.984	0.994	0.991	0.985	0.986	0.989	0.985	0.977	0.992	0.000	0.989	0.985	
Markets (B)	0.991	0.989	0.978	0.991	0.985	0.990	0.994	0.994	0.992	0.981	0.979	0.968	0.991	0.993	0.968	0.985	0.993	0.982	0.995	0.994	0.985	0.934	0.996	0.993	0.992	0.990	0.902	0.992	0.985	0.987	0.991	0.993	0.984	0.953	0.996	0.965	0.991	0.981	0.993	0.989	0.000	0.969
Religion (US)	0.980	0.992	0.980	0.982	0.983	0.986	0.986	0.988	0.988	0.992	0.989	0.986	0.991	0.986	0.964	0.979	0.984	0.974	0.987	0.991	0.982	0.990	0.992	0.983	0.989	0.986	0.985	0.989	0.988	0.983	0.989	0.985	0.990	0.971	0.987	0.958	0.982	0.960	0.986	0.985	0.969	0.000

Table B.5: MCS Distance between Node-Edge Samples (50%)

	Africa (W)	Canada (W)	Environment (Sc)	Mideast (W)	Soccer (Sp)	Animals (Sc)	Celebrity (E)	Europe (W)	Motor (Sp)	Software (T)	Apple (T)	Comm (T)	Fashion (E)	Movies (E)	Space (Sc)	Asia (W)	Crimes/Trials (US)	Football (Sp)	Music (E)	Tennis (S)	Australia (W)	Earnings (B)	Golf (Sp)	NCAA (Sp)	TV (E)	Baseball (Sp)	Economy (B)	Hockey (S)	Open source (T)	Weather (Sc)	Basketball (Sp)	Education (US)	Internet (T)	Personal (T)	Biotech (T)	Industry (E)	Latin (W)	Politics (US)	Books (E)	Energy (Sc)	Markets (B)	Religion (US)
Africa (W)	0.000	0.998	0.975	0.977	0.999	0.973	0.994	0.973	0.997	0.993	0.993	0.987	0.991	0.995	0.988	0.976	0.987	0.991	0.993	0.995	0.978	0.996	0.992	0.993	0.990	0.991	0.987	0.996	0.996	0.968	0.992	0.992	0.986	0.987	0.994	0.992	0.975	0.957	0.990	0.991	0.992	0.973
Canada (W)	0.998	0.000	0.988	0.998	0.994	0.984	0.980	0.992	0.998	0.992	0.990	0.988	0.974	0.992	0.982	0.998	0.996	0.980	0.978	1.000	0.982	0.996	0.996	0.990	0.988	0.994	0.980	0.994	0.992	0.992	0.990	0.992	0.978	0.994	0.992	0.984	0.998	0.982	0.988	0.984	0.970	0.956
Environment (Sc)	0.975	0.988	0.000	0.989	0.995	0.971	0.989	0.985	0.991	0.986	0.994	0.985	0.988	0.991	0.953	0.982	0.985	0.994	0.996	0.993	0.958	0.987	0.996	0.992	0.985	0.993	0.977	0.994	0.992	0.965	0.990	0.983	0.977	0.989	0.995	0.979	0.977	0.962	0.990	0.952	0.984	0.966
Mideast (W)	0.977	0.998	0.989	0.000	0.996	0.990	0.988	0.989	0.993	0.991	0.992	0.991	0.996	0.991	0.993	0.947	0.993	0.995	0.991	0.996	0.982	0.996	0.996	0.994	0.994	0.994	0.989	0.996	0.994	0.993	0.994	0.996	0.989	0.991	0.999	0.994	0.969	0.976	0.993	0.996	0.994	0.974
Soccer (Sp)	0.999	0.994	0.995	0.996	0.000	0.992	0.989	0.978	0.991	0.991	0.987	0.978	0.989	0.987	0.954	0.984	0.980	0.954	0.984	0.987	0.992	0.986	0.995	0.989	0.984	0.981	0.992	0.970	0.994	0.986	0.984	0.989	0.991	0.970	0.998	0.942	0.989	0.986	0.987	0.995	0.967	0.943
Animals (Sc)	0.973	0.984	0.971	0.990	0.992	0.000	0.991	0.991	0.997	0.992	0.993	0.990	0.991	0.993	0.982	0.975	0.987	0.993	0.995	0.997	0.951	0.995	0.994	0.993	0.992	0.992	0.986	0.993	0.993	0.989	0.990	0.996	0.985	0.991	0.980	0.991	0.986	0.986	0.988	0.992	0.989	0.986
Celebrity (E)	0.994	0.980	0.989	0.988	0.989	0.991	0.000	0.993	0.993	0.993	0.996	0.991	0.986	0.799	0.994	0.990	0.982	0.990	0.769	0.996	0.992	0.988	0.998	0.988	0.752	0.988	0.993	0.988	0.995	0.986	0.985	0.997	0.986	0.990	0.996	0.979	0.990	0.989	0.984	0.997	0.996	0.986
Europe (W)	0.973	0.992	0.985	0.989	0.978	0.991	0.993	0.000	0.994	0.992	0.993	0.995	0.975	0.993	0.995	0.984	0.979	0.996	0.996	0.992	0.983	0.999	0.995	0.996	0.998	0.995	0.990	0.993	0.996	0.993	0.996	0.997	0.993	0.995	0.995	0.991	0.985	0.981	0.992	0.993	0.996	0.989
Motor (Sp)	0.997	0.998	0.991	0.993	0.991	0.997	0.993	0.994	0.000	0.992	0.997	0.993	0.992	0.989	0.989	0.996	0.993	0.993	0.992	0.994	0.995	0.995	0.988	0.981	0.994	0.984	0.994	0.988	0.995	0.994	0.976	0.996	0.990	0.996	1.000	0.940	0.988	0.991	0.996	0.997	0.995	0.989
Software (T)	0.993	0.992	0.986	0.991	0.991	0.992	0.993	0.992	0.992	0.000	0.950	0.974	0.994	0.995	0.996	0.997	0.993	0.991	0.993	0.996	0.995	0.977	0.996	0.995	0.993	0.993	0.989	0.995	0.922	0.995	0.996	0.995	0.946	0.931	0.995	0.991	0.995	0.994	0.992	0.995	0.989	0.995
Apple (T)	0.993	0.990	0.994	0.992	0.987	0.993	0.996	0.993	0.997	0.950	0.000	0.967	0.994	0.990	0.997	0.996	0.995	0.992	0.992	0.998	0.996	0.970	0.999	0.996	0.993	0.992	0.986	0.993	0.958	0.994	0.994	0.992	0.978	0.935	0.993	0.985	0.994	0.992	0.995	0.996	0.986	0.989
Comm (T)	0.987	0.988	0.985	0.991	0.978	0.990	0.991	0.995	0.993	0.974	0.967	0.000	0.989	0.986	0.992	0.988	0.990	0.993	0.989	0.999	0.986	0.930	0.997	0.995	0.985	0.990	0.973	0.992	0.981	0.985	0.995	0.992	0.971	0.637	0.993	0.962	0.997	0.989	0.990	0.992	0.971	0.987
Fashion (E)	0.991	0.974	0.988	0.996	0.989	0.991	0.986	0.975	0.992	0.994	0.994	0.989	0.000	0.979	0.989	0.996	0.989	0.987	0.985	0.997	0.992	0.988	0.999	0.993	0.984	0.982	0.988	0.993	0.989	0.985	0.987	0.993	0.981	0.985	0.995	0.980	0.994	0.986	0.985	0.995	0.988	0.983
Movies (E)	0.995	0.992	0.991	0.991	0.987	0.993	0.799	0.993	0.989	0.995	0.990	0.986	0.979	0.000	0.994	0.993	0.990	0.993	0.876	0.995	0.991	0.987	0.999	0.983	0.809	0.981	0.997	0.987	0.994	0.989	0.986	0.993	0.985	0.992	0.998	0.925	0.995	0.993	0.980	0.998	0.997	0.989
Space (Sc)	0.988	0.982	0.953	0.993	0.954	0.982	0.994	0.995	0.989	0.996	0.997	0.992	0.989	0.994	0.000	0.993	0.946	0.984	0.996	0.997	0.992	0.997	0.999	0.996	0.998	0.997	0.991	0.995	0.997	0.988	0.994	0.996	0.995	0.973	0.997	0.957	0.994	0.992	0.997	0.989	0.967	0.957
Asia (W)	0.976	0.998	0.982	0.947	0.984	0.975	0.990	0.984	0.996	0.997	0.996	0.988	0.996	0.993	0.993	0.000	0.979	0.996	0.995	0.995	0.968	0.992	0.997	0.996	0.996	0.992	0.987	0.996	0.994	0.983	0.996	0.994	0.992	0.991	0.997	0.991	0.982	0.962	0.990	0.987	0.992	0.986
Crimes/Trials (US)	0.987	0.996	0.985	0.993	0.980	0.987	0.982	0.979	0.993	0.993	0.995	0.990	0.989	0.990	0.946	0.979	0.000	0.989	0.972	0.992	0.985	0.992	0.993	0.986	0.985	0.988	0.995	0.990	0.994	0.977	0.992	0.988	0.983	0.990	0.994	0.990	0.988	0.990	0.988	0.996	0.990	0.974
Football (Sp)	0.991	0.980	0.994	0.995	0.954	0.993	0.990	0.996	0.993	0.991	0.992	0.993	0.987	0.993	0.984	0.996	0.989	0.000	0.990	0.996	0.997	0.994	0.995	0.977	0.989	0.972	0.997	0.967	0.995	0.989	0.971	0.995	0.993	0.973	0.998	0.946	0.996	0.991	0.996	0.998	0.979	0.955
Music (E)	0.993	0.978	0.996	0.991	0.984	0.995	0.769	0.996	0.992	0.993	0.992	0.989	0.985	0.876	0.996	0.995	0.972	0.990	0.000	0.996	0.994	0.989	0.994	0.981	0.835	0.977	0.993	0.982	0.988	0.990	0.984	0.989	0.956	0.992	0.997	0.962	0.994	0.991	0.978	0.999	0.994	0.984
Tennis (S)	0.995	1.000	0.993	0.996	0.987	0.997	0.996	0.992	0.994	0.996	0.998	0.999	0.997	0.995	0.997	0.995	0.992	0.996	0.996	0.000	0.977	0.990	0.980	0.971	0.987	0.973	0.992	0.968	0.996	0.985	0.974	0.993	0.992	0.999	0.998	0.989	0.982	0.989	0.990	0.999	0.991	0.987
Australia (W)	0.978	0.982	0.958	0.982	0.992	0.951	0.992	0.983	0.995	0.995	0.996	0.986	0.992	0.991	0.992	0.968	0.985	0.997	0.994	0.977	0.000	0.986	0.996	0.993	0.990	0.994	0.992	0.996	0.992	0.982	0.993	0.992	0.979	0.985	0.996	0.995	0.987	0.955	0.986	0.982	0.986	0.983
Earnings (B)	0.996	0.996	0.987	0.996	0.986	0.995	0.988	0.999	0.995	0.977	0.970	0.930	0.988	0.987	0.997	0.992	0.992	0.994	0.989	0.990	0.986	0.000	0.997	0.996	0.981	0.991	0.952	0.991	0.978	0.983	0.990	0.990	0.979	0.953	0.994	0.978	0.998	0.981	0.982	0.987	0.910	0.984
Golf (Sp)	0.992	0.996	0.996	0.996	0.995	0.994	0.998	0.995	0.988	0.996	0.999	0.997	0.999	0.999	0.999	0.997	0.993	0.995	0.994	0.980	0.996	0.997	0.000	0.983	0.991	0.983	0.998	0.983	0.997	0.993	0.986	0.995	0.992	0.996	0.999	0.991	0.997	0.997	0.997	0.995	0.998	0.989
NCAA (Sp)	0.993	0.990	0.992	0.994	0.989	0.993	0.988	0.996	0.981	0.995	0.996	0.995	0.993	0.983	0.996	0.996	0.986	0.977	0.981	0.971	0.993	0.996	0.983	0.000	0.994	0.984	0.995	0.969	0.994	0.990	0.925	0.992	0.992	0.994	1.000	0.985	0.994	0.986	0.997	0.996	0.998	0.984
TV (E)	0.990	0.988	0.985	0.994	0.984	0.992	0.752	0.998	0.994	0.993	0.993	0.985	0.984	0.809	0.998	0.996	0.985	0.989	0.835	0.987	0.990	0.981	0.991	0.994	0.000	0.981	0.992	0.985	0.995	0.978	0.983	0.980	0.910	0.984	0.998	0.956	0.996	0.978	0.977	0.998	0.990	0.972
Baseball (Sp)	0.991	0.994	0.993	0.994	0.981	0.992	0.988	0.995	0.984	0.993	0.992	0.990	0.982	0.981	0.997	0.992	0.988	0.972	0.977	0.973	0.994	0.991	0.983	0.984	0.981	0.000	0.992	0.970	0.989	0.987	0.960	0.993	0.988	0.989	0.997	0.979	0.993	0.990	0.987	0.997	0.992	0.981
Economy (B)	0.987	0.980	0.977	0.989	0.992	0.986	0.993	0.990	0.994	0.989	0.986	0.973	0.988	0.997	0.991	0.987	0.995	0.997	0.993	0.992	0.992	0.952	0.998	0.995	0.992	0.992	0.000	0.994	0.993	0.983	0.990	0.993	0.986	0.981	0.994	0.983	0.990	0.959	0.988	0.983	0.887	0.981
Hockey (S)	0.996	0.994	0.994	0.996	0.970	0.993	0.988	0.993	0.988	0.995	0.993	0.992	0.993	0.987	0.995	0.996	0.990	0.967	0.982	0.968	0.996	0.991	0.983	0.969	0.985	0.970	0.994	0.000	0.990	0.986	0.936	0.992	0.987	0.990	1.000	0.986	0.994	0.990	0.983	0.998	0.991	0.984
Open source (T)	0.996	0.992	0.992	0.994	0.994	0.993	0.995	0.996	0.995	0.922	0.958	0.981	0.989	0.994	0.997	0.994	0.994	0.995	0.988	0.996	0.992	0.978	0.997	0.994	0.995	0.989	0.993	0.990	0.000	0.991	0.995	0.996	0.977	0.982	0.993	0.986	0.997	0.992	0.996	0.996	0.995	0.992
Weather (Sc)	0.968	0.992	0.965	0.993	0.986	0.989	0.986	0.993	0.994	0.995	0.994	0.985	0.985	0.989	0.988	0.983	0.977	0.989	0.990	0.985	0.982	0.983	0.993	0.990	0.978	0.987	0.983	0.986	0.991	0.000	0.987	0.983	0.992	0.989	0.997	0.989	0.986	0.937	0.991	0.992	0.994	0.979
Basketball (Sp)	0.992	0.990	0.990	0.994	0.984	0.990	0.985	0.996	0.976	0.996	0.994	0.995	0.987	0.986	0.994	0.996	0.992	0.971	0.984	0.974	0.993	0.990	0.986	0.925	0.983	0.960	0.990	0.936	0.995	0.987	0.000	0.993	0.986	0.993	0.999	0.989	0.995	0.991	0.990	0.998	0.991	0.992
Education (US)	0.992	0.992	0.983	0.996	0.989	0.996	0.997	0.997	0.996	0.995	0.992	0.992	0.993	0.993	0.996	0.994	0.988	0.995	0.989	0.993	0.992	0.990	0.995	0.992	0.980	0.993	0.993	0.992	0.996	0.983	0.993	0.000	0.990	0.987	0.997	0.984	0.992	0.937	0.984	0.990	0.990	0.973
Internet (T)	0.986	0.978	0.977	0.989	0.991	0.985	0.986	0.993	0.990	0.946	0.978	0.971	0.981	0.985	0.995	0.992	0.983	0.993	0.956	0.992	0.979	0.979	0.992	0.992	0.910	0.988	0.986	0.987	0.977	0.992	0.986	0.990	0.000	0.942	0.996	0.937	0.999	0.989	0.990	0.997	0.989	0.990
Personal (T)	0.987	0.994	0.989	0.991	0.970	0.991	0.990	0.995	0.996	0.931	0.935	0.637	0.985	0.992	0.973	0.991	0.990	0.973	0.992	0.999	0.985	0.953	0.996	0.994	0.984	0.989	0.981	0.990	0.982	0.989	0.993	0.987	0.942	0.000	0.993	0.926	0.996	0.991	0.988	0.992	0.935	0.957
Biotech (T)	0.994	0.992	0.995	0.999	0.998	0.980	0.996	0.995	1.000	0.995	0.993	0.993	0.995	0.998	0.997	0.997	0.994	0.998	0.997	0.998	0.996	0.994	0.999	1.000	0.998	0.997	0.994	1.000	0.993	0.997	0.999	0.997	0.996	0.993	0.000	0.991	0.997	0.988	0.995	0.981	0.991	0.961
Industry (E)	0.992	0.984	0.979	0.994	0.942	0.991	0.979	0.991	0.940	0.991	0.985	0.962	0.980	0.925	0.957	0.991	0.990	0.946	0.962	0.989	0.995	0.978	0.991	0.985	0.956	0.979	0.983	0.986	0.986	0.989	0.989	0.984	0.937	0.926	0.991	0.000	0.998	0.994	0.992	0.995	0.966	0.954
Latin (W)	0.975	0.998	0.977	0.969	0.989	0.986	0.990	0.985	0.988	0.995	0.994	0.997	0.994	0.995	0.994	0.982	0.988	0.996	0.994	0.982	0.987	0.998	0.997	0.994	0.996	0.993	0.990	0.994	0.997	0.986	0.995	0.992	0.999	0.996	0.997	0.998	0.000	0.977	0.975	0.995	0.989	0.974
Politics (US)	0.957	0.982	0.962	0.976	0.986	0.986	0.989	0.981	0.991	0.994	0.992	0.989	0.986	0.993	0.992	0.962	0.990	0.991	0.991	0.989	0.955	0.981	0.997	0.986	0.978	0.990	0.959	0.990	0.992	0.937	0.991	0.937	0.989	0.991	0.988	0.994	0.977	0.000	0.981	0.977	0.993	0.928
Books (E)	0.990	0.988	0.990	0.993	0.987	0.988	0.984	0.992	0.996	0.992	0.995	0.990	0.985	0.980	0.997	0.990	0.988	0.996	0.978	0.990	0.986	0.982	0.997	0.997	0.977	0.987	0.988	0.983	0.996	0.991	0.990	0.984	0.990	0.988	0.995	0.992	0.975	0.981	0.000	0.996	0.988	0.987
Energy (Sc)	0.991	0.984	0.952	0.996	0.995	0.992	0.997	0.993	0.997	0.995	0.996	0.992	0.995	0.998	0.989	0.987	0.996	0.998	0.999	0.999	0.982	0.987	0.995	0.996	0.998	0.997	0.983	0.998	0.996	0.992	0.998	0.990	0.997	0.992	0.981	0.995	0.995	0.977	0.996	0.000	0.984	0.973
Markets (B)	0.992	0.970	0.984	0.994	0.967	0.989	0.996	0.996	0.995	0.989	0.986	0.971	0.988	0.997	0.967	0.992	0.990	0.979	0.994	0.991	0.986	0.910	0.998	0.998	0.990	0.992	0.887	0.991	0.995	0.994	0.991	0.990	0.989	0.935	0.991	0.966	0.989	0.993	0.988	0.984	0.000	0.960
Religion (US)	0.973	0.956	0.966	0.974	0.943	0.986	0.986	0.989	0.989	0.995	0.989	0.987	0.983	0.989	0.957	0.986	0.974	0.955	0.984	0.987	0.983	0.984	0.989	0.984	0.972	0.981	0.981	0.984	0.992	0.979	0.992	0.973	0.990	0.957	0.961	0.954	0.974	0.928	0.987	0.973	0.960	0.000

Table B.6: SID Distance between Node-Edge Samples (50%)