Creating Social Networks to Improve Peer-to-Peer Networking

Andrew Fast, David Jensen, and Brian Neil Levine

Department of Computer Science
University of Massachusetts Amherst
140 Governors Drive, Amherst, MA 01003-9264
{afast, jensen, brian}@cs.umass.edu

ABSTRACT

We use knowledge discovery techniques to guide the creation of efficient overlay networks for peer-to-peer file sharing. An overlay network specifies the logical connections among peers in a network and is distinct from the physical connections of the network. It determines the order in which peers will be queried when a user is searching for a specific file. To better understand the role of the network overlay structure in the performance of peer-to-peer file sharing protocols, we compare several methods for creating overlay networks. We analyze the networks using data from a campus network for peer-to-peer file sharing that recorded anonymized data on 6,528 users sharing 291,925 music files over an 81-day period. We propose a novel protocol for overlay creation based on a model of user preference identified by latent-variable clustering with hierarchical Dirichlet processes (HDPs). Our simulations and empirical studies show that the clusters of songs created by HDPs effectively model user behavior and can be used to create desirable network overlays that outperform alternative approaches.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Measurement, Performance

Keywords

peer-to-peer networks, hierarchical dirichlet processes, social networks, distributed hash tables, overlay networks

KDD'05, August 21–24, 2005, Chicago, Illinois, USA.

1. INTRODUCTION

As peer-to-peer (P2P) file-sharing systems such as KaZaa and Gnutella increase in popularity, the efficiency of simple search methods, such as flooding, necessarily decreases. As the name implies, peers that utilize flooding search forward queries to all neighboring peers "flooding" the network with requests. Many researchers have attempted to increase efficiency with content-based overlay networks, including distributed hash tables (e.g., [1, 10]) and other semantic approaches (e.g., [4, 5, 12]). An overlay network specifies the logical connections between peers in a network and is distinct from the physical connections of that network. It determines the order in which peers are queried when a user is searching for a specific file.

In this paper, we present a new method for creating overlay networks that are based on a learned model of user preference and the musical styles of user libraries. Previous approaches depended on specific content already present in a user's library and provide no learned model to generalize the types of files users might prefer. By generalizing the files a user shares into a model of the types of files that a user prefers, we are able to build an overlay network connecting users who are likely to share files with each other. This allows us to create and capitalize on file locality specific to an individual user with particular preferences without relying on complex search methods or overly detailed user characteristics. We chose to identify styles (i.e., groups of files which people tend to prefer together) by clustering the files available in the network with hierarchical Dirichlet processes (HDPs). The only information needed to determine cluster assignments using an HDP is a list of filenames present in each users' shared library, information which is readily available in current P2P systems.

Our experiments and simulations show that by creating overlay networks based on social characteristics we are able to improve the performance of P2P networks. We demonstrate that clustering the MP3 audio files shared in an actual P2P network with HDPs captures our intuitive sense of musical styles and can be used to create an effective model of user download behavior. We then use that model to create overlay networks that connect users who prefer the same styles of music and demonstrate the overall effectiveness of those overlays when compared to random graphs, random cluster graphs, and direct file similarity graphs. We also demonstrate the utility of these new overlay networks when combined with a distributed hash table approach.

2. DATA DESCRIPTION

The data were collected from a campus network for P2P file sharing based on the OpenNap server. The data consist of records of all the files shared by and transferred between users during an 81-day period between February 28, 2003 and May 21, 2003. Users are uniquely identified by an anonymous MD5 hash. No personal information was collected during this study and users gave explicit consent to anonymous collection of the data. Files are uniquely identified by a filename and extension and are not limited to any particular filetype. In the raw data there were over 2 million distinct files. We chose to focus only on files with the MP3 extension, reducing the raw number of files to 466,221.

Rudimentary consolidation was performed by making all filenames lowercase, converting spaces and punctuation to dashes, and doing simple artist-name recognition. Most of the filenames contained some combination of the track name of the song, the song's artist, the track number and album name. The most common form of the filename was <artist>-<songname>.mp3. Using this information and some hand labeling, we were able to generate a list of the most prevalent artists in the database and use that information to help determine if two files should be consolidated. Through consolidation we reduced the number of files to 291,925. We did minimal consolidation on misspelled or alternate spellings of artist names or track names. By limiting the files to MP3s and performing simple name consolidation we were able to decrease the number of unique files by approximately 90% while only reducing the number of transfers and queries by 50% and the number of users by approximately 20%. Exact counts are shown in Figure 1.

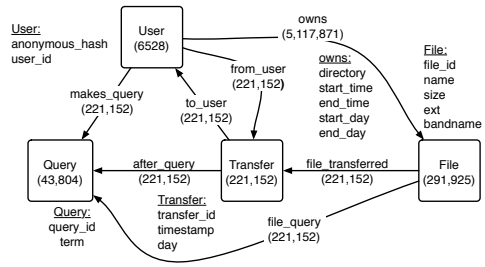

Figure 1: The P2P data schema showing counts of the objects and links after limiting the data to MP3 files and performing name consolidation.

User data were recorded twice daily at 12:00AM and 12:00 PM. Unfortunately, not all users were online when these snapshots of the network were taken. For example, there were 145 users who served files but never appeared in any snapshot. Transfers were recorded after a transaction was completed. To find a file, users queried a central database which returned an HTML page with links to files matching the query term. If a link was clicked, the time of the transaction, users involved, query term, and file transferred were all recorded. Chu et al. [3] provide a summary of statistics and trends present in the data.

3. IDENTIFYING STYLES OF FILES

Due to the inconsistency of information found in filenames, ID tags, industry labels, and music information sites on the web, it is difficult to determine the style or genre of a particular MP3 file, and more importantly, whether a user will download any given song. In place of labels and ID tags, we used clusters defined by a knowledge discovery algorithm to determine the styles of files in the system.

By representing user libraries as a document and files as terms, we can apply techniques from document clustering and topic detection to identify latent groups of files in user libraries. To find these latent groups, we chose to use a hierarchical Dirichlet process (HDP) [13] [14], a non-parametric extension to latent Dirichlet allocation [2], because it models each document as a mixture of latent topics. HDP is non-parametric in that the number of groups does not need to be provided *a priori*. Unlike text documents, where multiple occurances of words are meaningful, multiple instances of a single file appearing in a shared library should be disregarded. Also, the size of user libraries has a power law distribution (i.e., a small number of users have many files and many users have only a few files). Previously published experiments using HDPs cluster sets of documents with more uniform sizes. Despite these differences we were still able to use HDPs to identify desirable clusters, as described in Section 3.2.

3.1 Overview of the HDP Algorithm

An HDP is a non-parametric hierarchical Bayesian model involving multiple groups of data. The number of clusters is governed by a random variable that grows at a rate logarithmic in the number of data points. This model is generative and is based on the Dirichlet process mixture model. It is designed to generate groups of data where the individual items in each group are drawn from a mixture of distributions. A graphical model representation of an HDP is given in Figure 2.

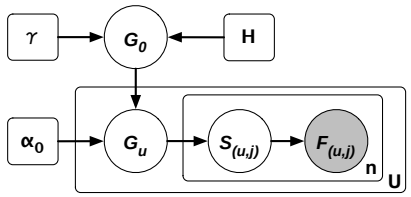

Figure 2: We model user libraries as a collection of files, F, labeled with a style descriptor, S. The distributions of the style parameters in user libraries is governed by a hierarchical Dirichlet process (HDP), the graphical model shown here.

We model U users each with a group or library of n files denoted by $\ell_u = (F_{(u,j)})_{j=1}^{|\ell|}$. We assume each file $F_{(u,j)}$ is drawn with conditional independence from a mixture model of genres with parameters set once for the group. Each user has a mix of musical tastes and each song in their library is taken from a style of music where the distribution of styles remains constant for each file in a user's library. Because each file is drawn independently, we can associate a genre or mixture component for each file. We use $S_{(u,j)}$ to denote the parameter specifying the genre for each file. In an HDP, each user is modeled with a Dirichlet process, $G_u \sim DP(\alpha_0, G_0)$, where the actual distribution over the parameters $S_{(u,j)}$ deviates from the base distribution G_0

with variability determined by some real number α_0. The distribution $G_0 \sim DP(\gamma, H)$ is also a Dirichlet process with base probability measure H and concentration parameter γ. The prior distribution for the parameters $(S_{(u,j)})_{j=1}^{U}$ is determined by the baseline H. It is important to note that the values of the parameters $S_{(u,j)}$ are shared between the users and within users' libraries.

3.2 Clustering Music Files

The HDP identified 99 clusters ranging in size from 239 files to 15 files. To reduce the size of our space, we clustered a limited set of 7888 files that were present in the first week of the data and appeared 3 or more times in the network. We assigned songs to their most probable cluster and used these clusters to define *styles* of groups of files . Representative styles are displayed in Table 1. While many of the clusters correspond to typical music industry genre labels (e.g., rock, hip hop, country, etc.), other clusters are best labeled with other categories. For example, Cluster 9 is a "popular songs" or "greatest hits" cluster. The cluster contains a broad range of popular artists and songs, including many classic artists such as Elvis and Van Morrison. Cluster 54 is dominated by female artists with no preference for a particular style or genre of music. Because of these types of clusters, we have chosen the term "style" instead of "genre" to describe the groups.

3.3 Cluster Evaluation

If we assume that styles are representative of true groups of files, then we would expect (1) songs from a given style to appear together in user libraries and (2) users to prefer songs from a small number of styles. For comparison, we also assigned files into 99 random clusters with the same precise probabilities of a file occurring in an HDP cluster. More than 80% of pairs of files drawn from the same HDP cluster co-occur in 1 or more user libraries. In contrast, approximately 80% of pairs of files drawn from random clusters of the same size do *not* co-occur in any user library. The histograms of these counts are shown in Figure 3. This verifies our expectations about the network and the correctness of the model.

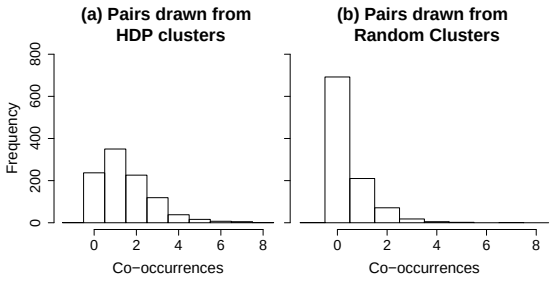

Figure 3: Co-occurrences of 1000 pairs of files in user libraries drawn from clusters and drawn at random. Pairs of songs drawn from HDP clusters co-occur many more times than pairs drawn from random clusters.

Figure 4 shows the distributions of the number of clusters per user. Figure 4(a) shows that for the majority of users 80% of their shared files can be described by only 20% of the HDP clusters. Most users, however, still own a small number of files from many clusters as is shown in Figure 4(b). Random clusters do not have the same descriptive power as the HDP clusters. These evaluations show that the HDP clusters match our expectations for successful clusters and we therefore use these clusters to build overlay networks that can connect users who prefer music files from the same clusters.

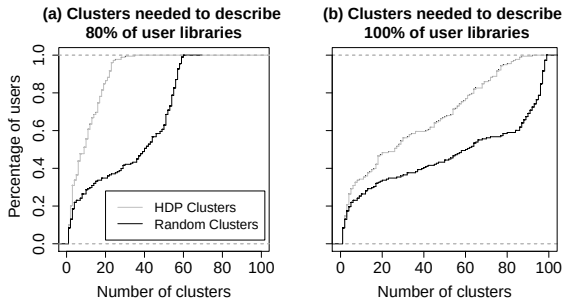

Figure 4: The distribution of clusters needed to describe user libraries. Fewer clusters per user mean that the clusters are more indicative of user tastes.

4. DESIGNING OVERLAY NETWORKS

Overlay networks specify the logical connections between users in a P2P network. Each user maintains a list of neighbors (or peers) whom they are able to contact. When a user wants to search for a file, they send a query to their neighbors, who pass it on to their neighbors and so on. These connections are easily represented as a graph. The original overlays for P2P networks were random graphs. Because no attempt was made to connect similar users, query performance varied from user to user depending on the type of users within a few hops. To introduce more consistency in the network, some content-based overlay networks been attempted (e.g., [4, 5, 12]). While these approaches have had moderate success, we believe that learned models of user behavior are necessary for major performance gains.

A plausible content-based alternative to random overlay networks is to build a network based on a measure of similarity between users' libraries. Unfortunately, this kind of direct file similarity does not capture important aspects of download behavior in a P2P network. Consider the pathological case. Imagine two users who both deeply enjoy listening to the music of the Rolling Stones. By coincidence, each of these two users owns exactly half the Rolling Stones catalog and do not share any files in common. They have zero songs in common but should still be linked together in the network based on the fact that they both like the Rolling Stones and would likely download many files from each other. At the other extreme, with direct file similarity two users with exactly the same library would be linked even though there would be very few transactions between these users. To balance these extremes, an efficient overlay network would connect users who share similar style preferences but do not already share many of the same files.

We propose creating overlay networks that connect users with similar distributions of the styles identified by the HDP clusters. Each user is identified by a vector denoting the probability of sharing a file of each style. We calculated this probability by counting the number of shared files in

Cluster 9 (Greatest Hits)	**Cluster 78** (Rap/Hip Hop)	**Cluster 54** (Female Artists)
enya-orinco-sail-away.mp3	tupac-i-ain't-mad-at-cha.mp3	tori-amos-spark.mp3
aerosmith-walk-this-way.mp3	ja-rule-furious.mp3	tiffany-i-think-we're-alone-now.mp3
van-morrison-brown-eyed-girl-1-.mp3	notorious-b.i.g.-big-poppa.mp3	britney-spears-baby-one-more-time.mp3
cranberries-linger.mp3	dmb-album-too-much.mp3	letters-to-cleo-here-and-now.mp3
bruce-springsteen-secret-garden.mp3	puff-daddy-victory.mp3	paula-abdul-straight-up.mp3
u2-sunday,-bloody-sunday.mp3	naughty-by-nature-jamboree.mp3	mariah-carey-fantasy.mp3
u2-stuck-in-a-moment.mp3	50-cent-21-questions.mp3	avril-lavigne-i'm-with-you.mp3
elvis-presley-don't-be-cruel.mp3	az-problems.mp3	cindy-lauper-time-after-time.mp3
bon-jovi-shot-through-the-heart.mp3	50-cent-in-da-club-rns.mp3	destiny's-child-survivor.mp3
avril-lavigne-complicated-1-.mp3	noreaga-superthug.mp3	shania-twain-any-man-of-mine.mp3
dave-matthews-band-satelite.mp3		no-doubt-simple-kind-of-life.mp3

Table 1: Top songs from selected clusters created by the HDP. (Cluster names added by author)

each style and dividing by the total library size. These calculations are described in Section 4.2. Because this is an abstraction over files, we can solve the problem experienced by the Rolling Stones fans by connecting users with many files of the same style even though they may not have many files in common. Also, we can factor out files in common and only connect users who have similar style distributions but not many files in common, solving the second pathological condition. In the next sections, we show that the styles found in user libraries and the styles of downloads by that user *are* similar and can be used to design efficient overlay networks.

4.1 Comparing downloads to libraries

We designed a test based on the chi-square statistic to determine whether the style distribution of user downloads are statistically similar to the style distribution of their libraries. First, we determined the background probability of a song being drawn from a given style based on the style distributions of the entire network. This background probability is calculated in Equation 1.

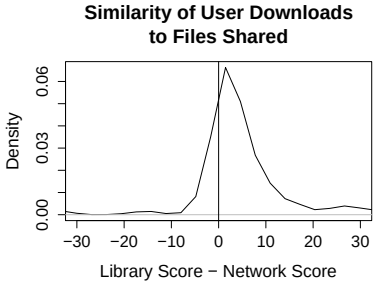

Figure 5: The distribution of similarity scores from Day 52 - 81 of the P2P data. Positive scores indicate a user's downloads are more like his/her library than the background network distribution.

$$P_b(s_i) = \frac{|songs\ in\ style_i|}{|songs\ present\ in\ network|} \quad (1)$$

We can calculate a similar probability for a user sharing a song in a given style.

$$P_u(s_i) = \frac{|songs\ shared\ in\ style_i|}{|songs\ shared\ by\ user\ u|} \quad (2)$$

Given a user's downloads, we can calculate the number of expected songs downloaded in each cluster by a user for both the background probability and the library probability.

$$E_u(s_i) = P_u(s_i) \cdot |downloads_u| \quad (3)$$
$$E_b(s_i) = P_b(s_i) \cdot |downloads_u| \quad (4)$$

Using these expected values, we can calculate two chi-square statistics to determine how similar a user's downloads are to the background style distributions and to their shared library distributions.

$$\chi^2_z = \sum_{i=1}^{|styles|} \frac{(|downloads_{s_i}| - E_z(s_i))^2}{E_z(s_i)} \quad , z \in \{u, b\} \quad (5)$$

Using the difference between these two statistics, we can determine if users are more like the network or more like their libraries. Figures 5 and 6 show the distributions of these statistics in the data. Because the majority of the non-zero scores are positive, we can conclude that users tend to download in proportion to the styles present in their shared libraries. The negative scores are problematic, however, as this indicates there are some users whose downloads are much more like the overall network and less like their own libraries. We explored this phenomenon by comparing the number of files shared and the number of downloads for each user. As is evident in Figure 6, the users with negative scores tend to download proportionally more songs than they share compared to the rest of the population. These users, called *freeloaders*, abuse the network by downloading many files without sharing those files and allowing other users to download files from them. This is evident in Figure 6. Because we would like to discourage freeloading, we will not consider freeloaders when designing our networks.

4.2 Connecting users with similar styles

Because users' downloads are similar to their libraries we can design an overlay network to connect users to sharers most likely to satisfy the anticipated queries of the downloader, making the music they prefer easier to find. We define the expected number of files that a sharer provides to a downloader as

$$E(u, d) = \sum_{i=1}^{|styles|} P_d(s_i)(|S_u(s_i)|) \quad (6)$$

where $P_d(s_i)$ is the probability of style i being downloaded by downloader d and $S_u(s_i)$ is the set of songs shared by user

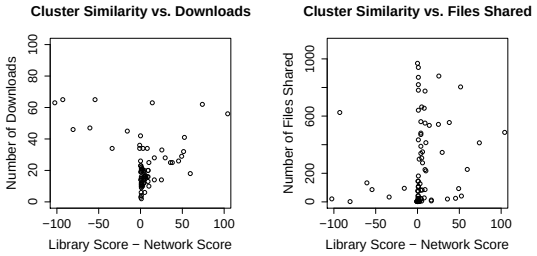

Figure 6: Comparison of downloads and files shared for a given style distribution score. Users with positive scores download files from the same styles that are present in their libraries. Users with negative scores are primarily freeloaders who download many times without making those files available to the network. The x-axis is the same in both (a) and (b) and there is a 1:1 correspondence between the points.

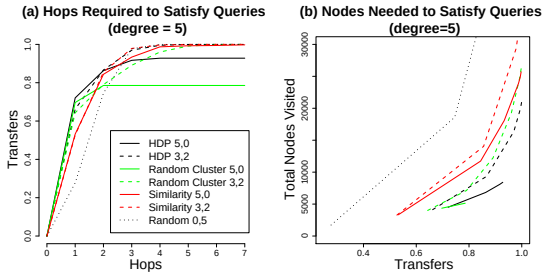

Figure 7: Performance of network overlays on 1250 transfers from Days 22-51 of the P2P data with users connecting to 5 other users. (a) Number of hops needed to satisfy queries. Hops are measured by the shortest path in the overlay network (b) Nodes visited if the search stopped after satisfying the query. Totals are averages over 10 runs.

u in style i not already owned by d. For each downloader we can rank every other user based on the expected number of new songs they might provide. As desired, users who share many files will likely have a large number of expected downloads for other users. However, having too many users connecting to a single other user causes an unbalanced distribution of work among all of the users. Using these ranked lists, we can create overlay networks with logical connections between the set of users and the top n other users in their ranked list. It is also possible to consider a hybrid approach where given a degree limit, l, a user selects k users from their ranked list and $l - k$ additional random links. This hybrid approach increases the connectivity of the resulting graph and leads to some important performance trade-offs, as described in the next section.

5. OVERLAY EVALUATION

We compared four different types of overlay networks: (1) networks using HDP styles; (2) networks using random styles; (3) networks using direct file similarity; and (4) random networks. To avoid edge effects and other anomalies, we analyzed a 30 day period from the middle of the data. We examined how performance was affected by the number of connections to other users (i.e., out degree) and the number of random connections. To better understand the effect of network size on performance, we analyzed 1, 2, 3, and 4-week samples from the original 30 days. The actual file downloads recorded in each sample time period were replayed over a simulated overlay network.

For each of the four types of overlay networks, we considered out degrees for each user ranging between 3 and 10. Each user was allowed the same number of connections. Users were connected to the top users in their ranked list for each of the non-random methods. Experiments using the hybrid approach described above varied the number of random links between 0 and the out degree. For example, if a user was allowed 5 outgoing connections, we simulated networks with between 0 and 5 random links.

As the networks increase in size and in the number of attempted queries, HDP begins to outperform the other approaches. As shown in Figure 7(a), the overlay networks based on the HDP styles satisfy more queries within one hop than than the equivalent random styles, similarity graph and random graph of the same degree. After one hop, the other approaches begin to catch up. In Figure 7, overlays followed by 3,2 represent a graph with 3 links chosen from the cluster and 2 random links. The best overall strategy with degree 5 is the hybrid HDP. After 2 hops, it performs equivalently to Similarity 3,2, but due to larger number of queries satisfied in one hop, the hybrid HDP approach bothers fewer users overall. Figure 7(b) demonstrates the total number of users needed to satisfy all the queries in days 22-51, if we were able to stop the search process at the level that satisfies the query. As one might expect, satisfying queries in a fewer number of hops causes an exponential reduction in the number of users queried. Even more difficult queries requiring a larger number of hops using the HDP styles never bother more users than the other overlay networks.

There are two factors that lead to increased performance of a single hop in the HDP and random style approaches. First, the HDP approach is attempting to connect users with similar music preferences. If the approach is working, then users are likely to find files they wish to download within a smaller number of hops than other approaches. Second, both the HDP approach and the random style approach favor connections to users with many shared files. This makes a large number of files available within a very few number of hops. If the requested file is *not* shared in one of these large libraries, then it may very difficult or even impossible to search the entire network for that file.

The hybrid approach is designed to counteract imbalanced work loads and the difficulties of finding rare songs. By allowing a small number of random links, the overall connectivity of the network increases as users are randomly connected to other users regardless of preference. This causes a small decrease in the number of queries satisfied within a single hop in exchange for satisfying many more of the queries for rare songs. The intuitions of small world network could explain the results of the hybrid approaches shown in Figure 7. According to Watts and Strogatz [15], nodes in small world networks are connected to many nodes within their cluster with a few long range or random links connecting the clusters. This also suggests an alternative method for searching in overlay networks. By maintaining multiple sets of connections, it would be possible to first search just the HDP style connections one hop away, and then, if the file

	Average Work	P(success)
Random	3	0.16
Random Cluster	3	0.585
HDP	3	0.64
Rand. Combo	8.07	1
DHT	6.04	1
Rand. Cluster Combo	5.50	1
HDP Combo	5.18*	1

Table 2: Summary of performance on 1051 queries from day 22 through day 51. HDP, Random, and Random Cluster results indicate success within a single hop. An asterisk (*) denotes significant improvements from the DHT. ($p \leq 0.02$)

isn't found, query a set of random connections. This approach has the benefits from the availability of large shared libraries without sacrificing ability to find rare files.

Recently, Loo et al. [10] described an approach for querying P2P networks that combines a distributed hash table (DHT) with a pre-existing P2P network. Because DHTs only utilize $\mathcal{O}(\log(n))$ nodes per query, where n is the number of users, rare files are easily found without the exponential work typical with flooding search. For popular queries on sufficiently large networks, however, DHTs are outperformed by P2P networks with random overlays. Due to the large increase in single hop performance using the HDP overlays, we are able to demonstrate significant improvements using the combination of an HDP overlay (out degree=3) and a DHT. These results are summarized in Table 2.

6. RELATED WORK

We chose HDPs to model musical styles. HDPs come from a family of soft-clustering techniques for topic detection in documents. The first of these approaches, probabilistic latent semantic indexing (pLSI), [7], has some difficulties with the generative semantics of the model making it very difficult to apply the model to new data. Latent Dirichlet allocation (LDA) [2] was designed to correct the generative semantics of the pLSI model and provide a more formal statistical model. HDPs were designed to be a hierarchical version of LDA that removed the requirement of specifying the number of latent topics *a priori*. Lavrenko presents an alternative approach to topic detection based on kernels [9]. He claims that HDPs and LDA are not desirable because they tend to lump outliers into existing clusters rather than creating new clusters. We experienced this with classical MP3 files; however, the amount of traffic due to these files was negligible when compared to the entire network.

Newman provides an overview of work analyzing graph structure and an understanding of how structure influences the function of the graph [11]. The work of Domingos and Richardson [6] and Kempe, Kleinberg and Tardos [8] provide insight into how people can be placed in a social network to maximize the influence they have on their surrounding neighbors. While our work does not seek to provide recommendations for files, these approaches could be used to determine how central a particular user should be in the network. This could be used to create overlay networks that account for popularity trends of files in the system by placing users sharing popular files at the center of the network.

7. ACKNOWLEDGMENTS

This research is supported by NSF and DARPA under contract numbers IIS0326249 and HR0011-04-1-0013. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of NSF, DARPA, or the U.S. Government.

8. REFERENCES

[1] *The Chord Project , http://www.pdos.lcs.mit.edu/chord/*, 2004.

[2] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, (3):993–1022, January 2003.

[3] J. Chu, K. Labonte, and B. N. Levine. Evaluating the use of Chord with real-world peer-to-peer traces. May 2003.

[4] E. Cohen, A. Fiat, and H. Kaplan. Associative search in peer to peer networks: Harnessing latent semantics. In *Proceedings of the IEEE INFOCOM'03 Conference*, 2003.

[5] A. Crespo and H. Garcia-Molina. Semantic overlay networks for p2p systems. Technical report, Stanford University, 2003.

[6] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining*, pages 57–66, 2001.

[7] T. Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, pages 50–57, Berkeley, California, August 1999.

[8] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of ACM SIGKDD 2003*.

[9] V. Lavrenko. *A Generative Theory Of Relevance*. PhD thesis, University of Massacusetts, September 2004.

[10] B.T. Loo, J.M. Hellerstein, R. Huebsch, S. Shenker, and I. Stoica. Enhancing P2P file-sharing with an internet-scale query processor. In *VLDB*, 2004.

[11] M.E.J. Newman. The structure and function of networks. *SIAM Review*, (45):167–256, 2003.

[12] K. Sripanidkulchai, B. Maggs, and H. Zhang. Efficient content location using interest-based locality in peer-to-peer systems. In *Proceedings of the IEEE INFOCOM'03 Conference*, 2003.

[13] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical Dirichlet processes. Technical Report 653, Department of Statistics, University of California at Berkeley, 2004.

[14] Y.W. Teh, M.I. Jordan, M.J. Beal, and D.M. Blei. Hierarchical Dirichlet processes. In *Advances in Neural Information Processing Systems*, volume 17, 2004.

[15] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.