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 the graph characteristics and social network of users generated by 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.

1. INTRODUCTION

As peer-to-peer 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 utilizing flooding search forward every query to all neighboring peers “flooding” the network with requests. Many researchers have attempted to increase efficiency with improved search techniques (e.g., distributed hash tables [1],[10]) and new algorithms for encoding information in overlay networks (e.g., [11],[9]). An overlay network specifies the logical connections between 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. These overlay networks do not consider the content available in the network when designing the network topology. Instead they concentrate on bandwidth and peer availability characteristics. Previous studies [5] have also shown that capitalizing on content locality in peer-to-peer (P2P) file-sharing networks is critical for reducing bandwidth consumption.

In this paper, we present a new method for creating overlay networks that is based on a model of user preference and the content of user libraries. 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 creating overlay networks based on social characteristics are able to improve the performance of peer-to-peer networks. We demonstrate that clustering the mp3 audio files shared in an actual peer-to-peer network with HDPs capture intuitive musical styles and can be used to create an effective model of user download behavior. We then use that model to create overlay networks which 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.

2. DATA DESCRIPTION

The data were collected from a campus network for peer-to-peer file sharing based on the OpenNap server. The data consists 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. There were 476,388 transfers and 7,886 users recorded in the time period under study. 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 names 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. Prior to consolidation there were 466,221 unique mp3 files; after consolidation there were 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.

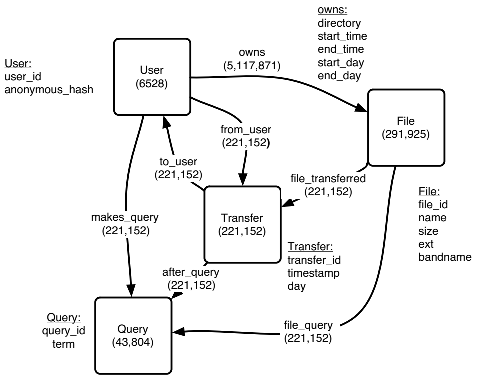


Figure 1: The P2P data schema showing counts of the objects and links after limiting the data to mp3 files and performing name consolidation.

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

Despite many attempts from the music industry and the music-loving public, it is still difficult to group music files into meaningful styles. On-line music information sites and music labels have attempted to identify groups of music by assigning each artist a particular genre. Artists are labeled by their dominant genre and individual songs by an artist are labeled with that genre as well regardless of the style or overall sound of that particular song.

The files in a peer-to-peer network do not have much reliable information attached to them. Filenames vary from user to user depending on the preferences of the users and the source of the file. Although many mp3 files contain ID tags embedded in the file, these values, if any, are user generated and are not reliable. 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 clusters or latent groups of files in user libraries. Documents are a point in a vector space where the vectors are unique terms appearing in the data. There are a number of popular approaches that limit documents to a single topic (e.g., k-means clustering). We chose hierarchical Dirichlet processes (HDPs) [13] [14], a non-parametric extension to latent Dirichlet allocation [2], because it models each document as a mixture of latent topics. It is non-parametric in that the number of groups does not need to be provided *a priori*. There are some key differences, however. Unlike documents where words can appear multiple times in a single document, users only share a single file once. We consolidated multiple instances of a single file appearing in a shared library. There is no need for duplicates of a song. 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). The published experiments of HDP are on document corpora of a much more uniform size. 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.

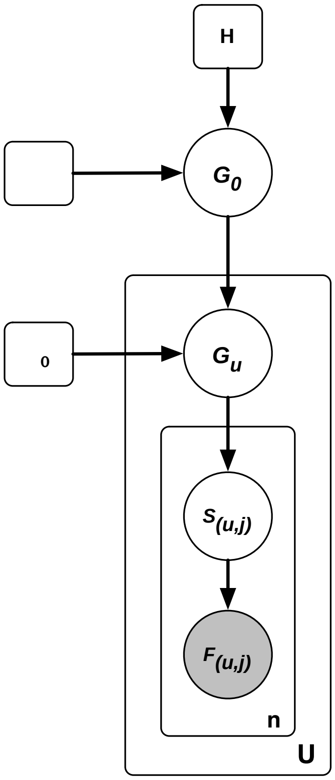


Figure 2: We model user libraries as a collection of files, F, labeled with a style descriptor, S. The distributions of the style parameters in user libraries is governed by a hierarchical Dirichlet process (HDP), the graphical model shown here.

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. Since 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 $(S_{(i,j)})_{j=1}^{U}$ is determined by the baseline H. It is important to note that the values the parameters $S_{(u,j)}$ are shared between the users and within users' libraries.

3.2 Clustering Music Files

The HDP identified 99 clusters in ranging in size from 239 files to 15 files. To reduce the dimensionality of our vector space, we only considered files present in the first week of the data occurring 3 or more times in users' libraries. This reduction left 7888 files to be clustered. We took all of the songs occurring in a cluster and assigned them to a style denoted by that cluster identification number. 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, heavy metal 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. A prominent exception is classical music. None of the clusters contain predominantly classical music. As might be expected from data collected on a university campus, there were few classical files shared in the network. These files were grouped in with other styles because they did not appear together often enough to generate their own style. After scanning Table 1, the song clusters appear to make intuitive sense.

3.3 Cluster Evaluation

If we assume that the 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. The histograms comparing these counts are shown in Figure 3. More than 80% of pairs of files drawn from the same cluster 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* occur together in any user library. This verifies our hypotheses about the network and the correctness of the model.

Figures 4 and 5 show the distributions of the number of clusters per user and the number of users per cluster, respectively. Figure 4(a) shows that for the majority of users roughly 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(c). Random clusters do not have the same descriptive power as the HDP clusters. Figure 5(a) shows that users are evenly distributed into clusters. The evaluations show that the HDP clusters match our assumptions about successful clusters and we may be able to use the clusters to build overlay networks which can connect users who prefer music files from the same clusters.

Cluster 9 (Greatest Hits)	**Cluster 78** (Rap/ Hip Hop)
enya-orinco-sail-away.mp3	tupac-i-ain't-mad-at-cha.mp3
meatloaf-paradise-by-the-dashboard-light.mp3	ja-rule-furious.mp3
van-morrison-brown-eyed-girl-1-.mp3	notorious-b.i.g.-big-poppa.mp3
cranberries-linger.mp3	dmb-album-too-much.mp3
bruce-springsteen-secret-garden.mp3	puff-daddy-victory.mp3
u2-sunday,-bloody-sunday.mp3	naughty-by-nature-jamboree.mp3
u2-stuck-in-a-moment.mp3	50-cent-21-questions-feat-nate-dogg-rns.mp3
elvis-presley-don't-be-cruel.mp3	az-problems.mp3
bon-jovi-shot-through-the-heart.mp3	50-cent-in-da-club-rns.mp3
avril-lavigne-complicated-1-.mp3	noreaga-superthug.mp3
dave-matthews-band-satelite.mp3	
sixpence-none-the-richer-kiss-me.mp3	
35-aerosmith-walk-this-way.mp3	

Cluster 54 (Female Artists)	**Cluster 17** (Punk)
tori-amos-spark.mp3	sleater-kinney-07-combat-rock.mp3
tiffany-i-think-we're-alone-now-1-.mp3	beastie-boys-11-and-me.mp3
britney-spears-baby-one-more-time.mp3	the-clash-03-jimmy-jazz.mp3
letters-to-cleo-here-and-now.mp3	green-day-paper-lanterns.mp3
paula-abdul-straight-up.mp3	sleater-kinney-03-turn-it-on.mp3
mariah-carey-fantasy.mp3	05-beastie-boys-time-for-livin'.mp3
melissa-etheridge-come-to-my-window-1-.mp3	the-clash-09-clampdown.mp3
cindy-lauper-time-after-time.mp3	sonic-youth-06-plastic-sun.mp3
avril-lavigne-i'm-with-you.mp3	beck-15-painted-eyelids.mp3
no-doubt-return-of-saturn-02-simple-kind-of-life.mp3	beastie-boys-02-the-move.mp3
destiny's-child-survivor.mp3	the-clash-18-revolution-rock.mp3
shania-twain-any-man-of-mine.mp3	

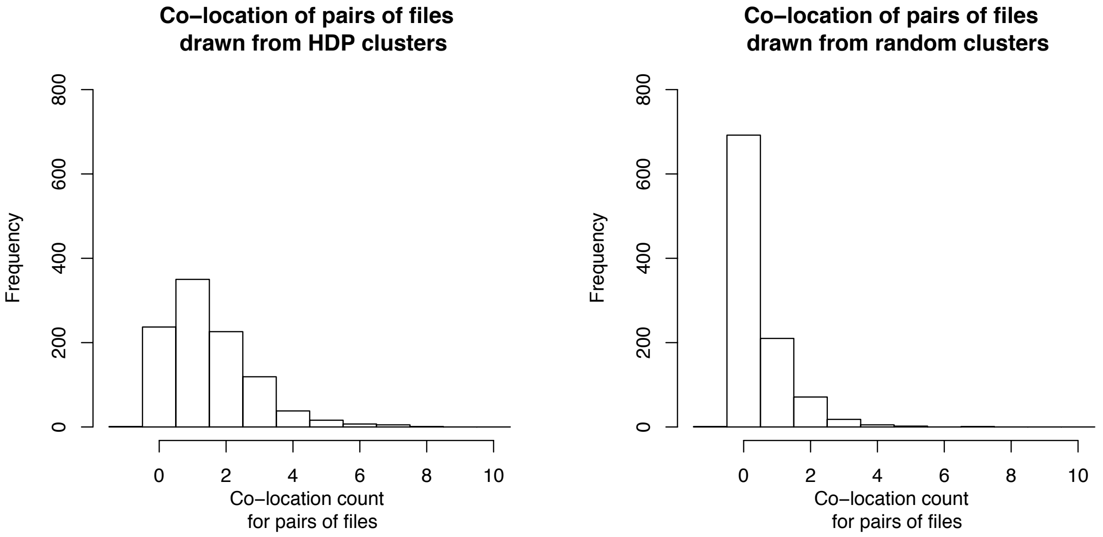
Table 1: Top songs from selected clusters created by the HDP. (Cluster names added by author)

4. DESIGNING OVERLAY NETWORKS

Overlay networks specify the logical connection between users in a P2P network. These networks can be represented as a graph with the set of connections between users as edges overlaid onto the users represented by vertices. Each users maintains a list of neighbors (or peers) who they are able to contact. When a user would like to query for a file, they send the query to their neighbors, who pass it on to their neighbors and so on. The original overlays for P2P networks were random graphs. Since 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. Some approaches based on bandwidth and availability have been attempted to introduce more consistency in the network (e.g., [11] [9]). While these approaches have had moderate success, we believe that content-based overlay networks are necessary for major improvements in query performance.

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 peer-to-peer 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. In fact, they should have a high probability of being linked as they enjoy the same style of music 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 need to 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 files in each style and dividing by the total library size. These calculations are described in Section 4.2. Since 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. In the same way, 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. For this approach to work, the styles of the shared files and downloads for users must be similar in some



(a)

(b)

Figure 3: 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.

way. Before considering the specifics of designing overlay networks, in the next section we demonstrate that the styles found in user libraries and styles of downloads by that user *are* similar.

4.1 Comparing downloads to libraries

Connecting users based on the distribution of styles in their shared libraries is only useful if the users also download files from the same style distribution. 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 cluster distributions of the entire network. This background probability is calculated in Equation 1.

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

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

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

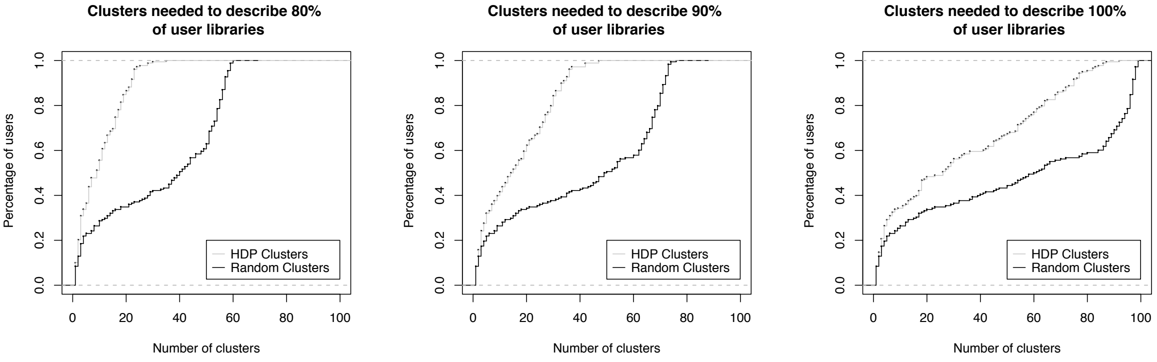
Given a users' downloads, we can calculate the number of expected songs downloaded in each cluster by a user for both the background probability and the library probability.

$$\begin{align} E_l(c_i) &= P_u(s_i) \cdot |downloads_u| \tag{3} \\ E_b(c_i) &= P_b(s_i) \cdot |downloads_u| \tag{4} \end{align}$$

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

$$\chi^2 = \sum_{i=1}^{|clusters|} \frac{(|downloads_{c_i}| - E(c_i))^2}{E(c_i)} \tag{5}$$

Using the difference between these two statistics we can determine if users are more like the network or more like their libraries. Figures 6 and 7 show the distributions of these statistics in the data. Since the majority of the non-zero scores are positive (i.e., library statistics are larger than background statistics) , we would like to conclude that users tend to download in proportion to the styles present in their shared libraries. The highly negative scores are problematic, however, as there are users who 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 7, the users that are more like the network tend to download proportionally more songs than they share compared to the rest of the population. See especially the data points



(a)

(b)

(c)

Figure 4: The distribution of clusters needed to describe user libraries. The fewer clusters needed to explain users indicates that the clusters are more indicative of user tastes.

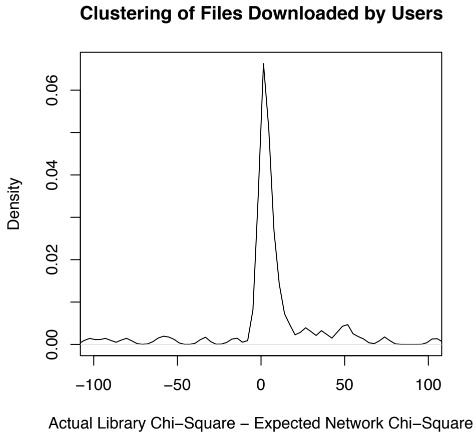


Figure 6: The distribution of similarity scores from Day 52 - 81 of the P2P data. Positive scores indicate a user downloads are more like his/her library than the background network distribution.

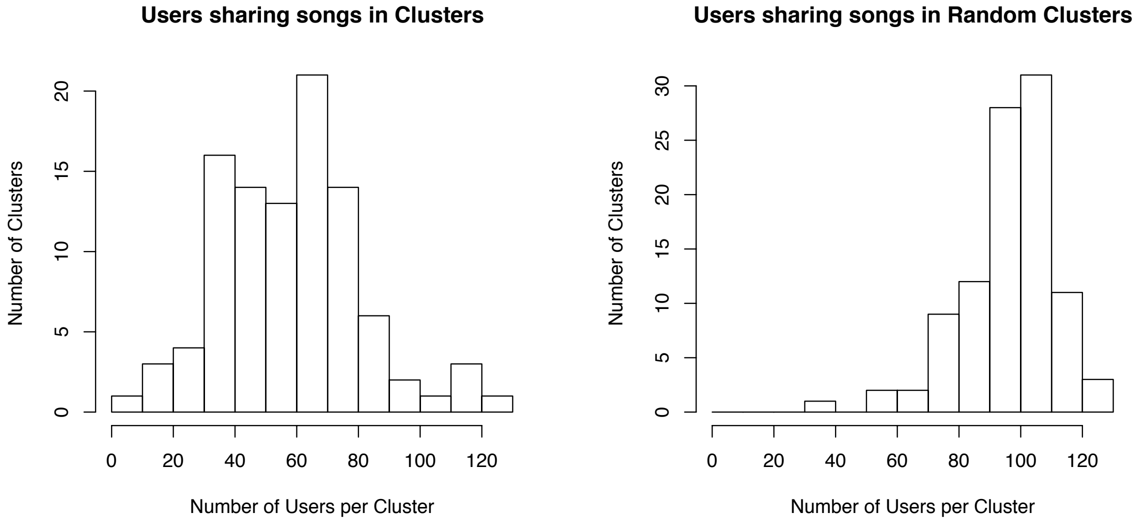
with negative scores. Each of these points correspond to a point with the same style score in the other figure. The users with negative scores, called *freeloaders*, abuse the network by downloading many files without sharing those files and allowing other users to download files from them. If they were actually taking part in the network, then the files downloaded by freeloaders should be added to their library but Figure 7 shows that the users with negative scores download many files but share very few. Since freeloaders do not contribute to the network, we will exclude them from future consideration and not incorporate them into our overlay networks. After excluding the freeloaders, the downloads by the remaining users are consistent with the style distributions of their libraries.

4.2 Connecting users with similar styles

Since users' downloads tend to be similar to their library we can design an overlay network to connect users who share files from the same styles, bringing the music they prefer closer in the network and therefore easier to find. We use the style distributions of each downloader to connect to sharers most likely to satisfy the anticipated queries of the downloader. We define the expected number of files that a sharer provides to a downloader as

$$E(u,d) = \sum_{i=1}^{|clusters|} P_{d_i}(c_i)(|Sc_i(u_i)|) \qquad (6)$$

where $P_{d_i}(c_i)$ is the probability of cluster i being downloaded by downloader d and $S_{c_u}(u_i)$ is the set of songs shared by user u in cluster 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. Users who share many files will likely have a large number of expected downloads for other users. This is a desired effect because users with many files are also more likely to be able satisfy queries from many 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, d, a user selects k users from their ranked list and $k - d$ additional random links. This hybrid approach increases the connectivity of the resulting graph and leads to some important performance tradeoffs, as described in the next section.



(a)

(b)

Figure 5: Users sharing songs in clusters. The majority of HDP clusters (a) contain files owned by between 30 and 80 users. In contrast, files in random clusters (b) are owned by between 80 and 110 users. This indicates that HDP clusters are better for discriminating users based on musical taste.

5. OVERLAY EVALUATION

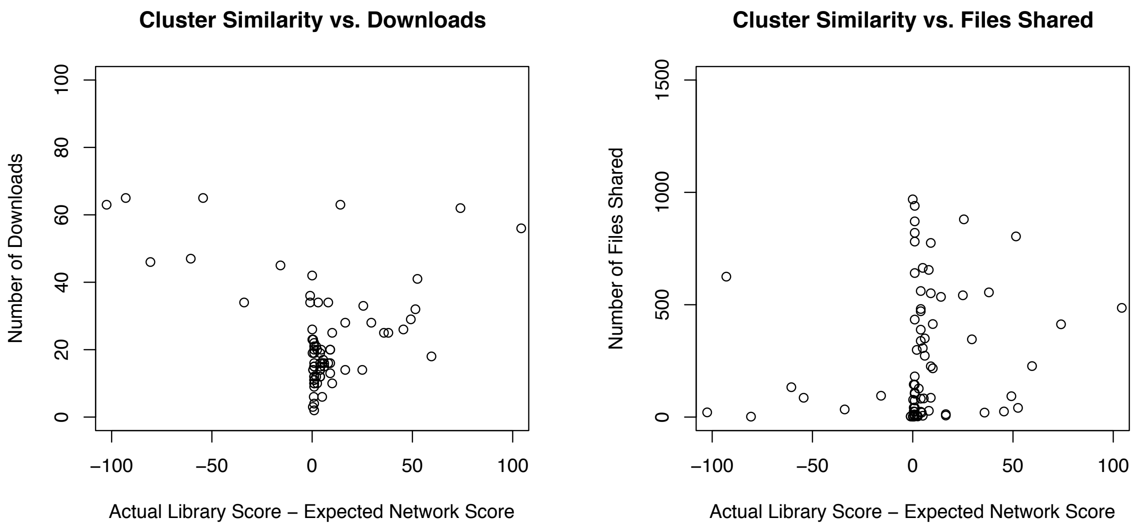
We compared four different types of overlay networks: 1) networks using HDP styles; 2) networks using Random styles; 3) networks using direct file similarity; and 4) random networks. To avoid any edge effects and other anomalies (e.g., spring break), we analyzed a 30 day period from the middle of the recorded data. We examined how performance was affected by the number of connections allowed to other users (i.e., out degree) and the number of random connections allowed. 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 and were connected to the top users in their ranked list for each of the non-random methods (i.e., HDP styles, random styles, and similarity). For each of the out degrees, the hybrid approach mentioned above was also considered by varying the number of random links allowed. For example, if a user was allowed 5 outgoing connections, we simulated networks with between 0 and 4 random links.

As the networks grow in the number of users and the number of attempted queries, HDP begins to outperform the other approaches. As shown in Figure 8(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. 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. (See Figure 8(b)).

There are two factors which 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 that file.

Favoring users with large shared libraries causes a skew in the amount of work performed by each user. One of the ideals of P2P networks is sharing work evenly among all of the peers. By favoring large libraries, the two overlay network approaches force a few users to handle many more queries than the average user. Figure 9 shows the number of connections to users with large li-



(a) (b)

Figure 7: Comparison of downloads and files shared for a given style distribution score. Users with positive scores download files from the same clusters that are present in their shared libraries. Users with negative scores are primarily freeloaders who download many times without making those files available to the rest of the network. The x-axis is the same in both (a) and (b) and there is a 1:1 correspondence between the points.

braries. The HDP style approach spreads the work over a larger number of users when compared to the random style approach. The random approach causes almost all the users to connect to the 4 users with the largest libraries.

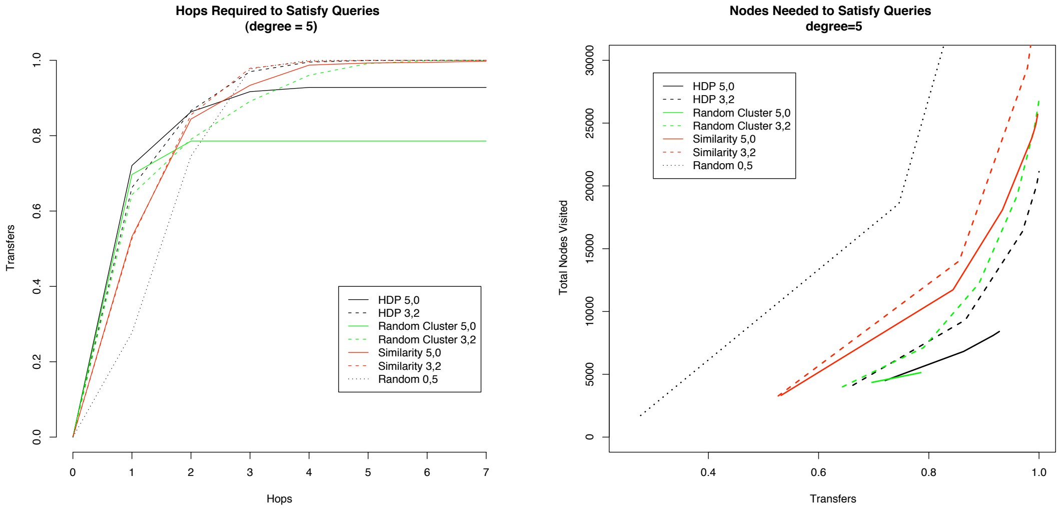
The hybrid approach is designed to offset some of the imbalances in work loads and the difficulties of finding rare songs. By allowing a small number of random links, the connectivity of the network increases as users are randomly connected to other users regardless of preference. This causes a small decrease in the number of queries satisfied within a single hop in exchange for satisfying many more of the queries for rare songs. This follows the intuition of Watts and Strogatz in their work on small world networks [15]. According to Watts and Strogatz, 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 type of structure would explain the results of the hybrid approach shown in Figure 8(a).

Figure 8(a) 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 isn't found, query a set of random connections. This approach has the benefits from the availability of large shared libraries with sacrificing ability to find rare files.

Another method of evaluating the overlay networks is counting the number of users queried to satisfy a particular query. Figure 8(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 the more difficult queries requiring a larger number of hops using the HDP styles never bother more users than what the other overlay networks require.

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), [6], 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 which removed the requirement of specifying the number of latent topics *a priori*. Lavrenko presents an alternative approach to topic detection based on kernels [8]. 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 phe-



(a) (b)

Figure 8: Performance of the network on 1250 transfers from Days 22-51 of the P2P data with users connecting to 5 other users. HDP 3,2 represents a graph with 3 links chosen from the cluster and 2 random links. (a) Number of Hops needed to satisfy queries. Hops are measured by the shortest path in the graph created by the overlay network (b) Nodes visited if the search stopped after satisfying the query. Totals reported are averages of 10 runs.

nomenon 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 understand how structure influences the function of the graph [12]. The work of Domingos and Richardson [4] and Kempe, Kleinberg and Tardos [7] 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 of 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. FUTURE WORK

There are a number of improvements to the algorithm that we would like to explore next. In our current work, the HDP cluster approach results in a few peers with a very high number of users connected to them. It would be interesting to incorporate some load balancing techniques such as a cap on the number of users allowed to connect to a peer. Currently, the HDP model is not decentralized and does not easily extend to allow the additions of new songs to the clusters. To make this a widely used system we would need to define a suitable procedure for allowing new users to successfully enter the system and a useful method for allowing freeloaders to participate in the system without drowning the legitimate users in excess traffic as well as developed a new online HDP algorithm which is able to assign files to clusters as they become prominent in the network.

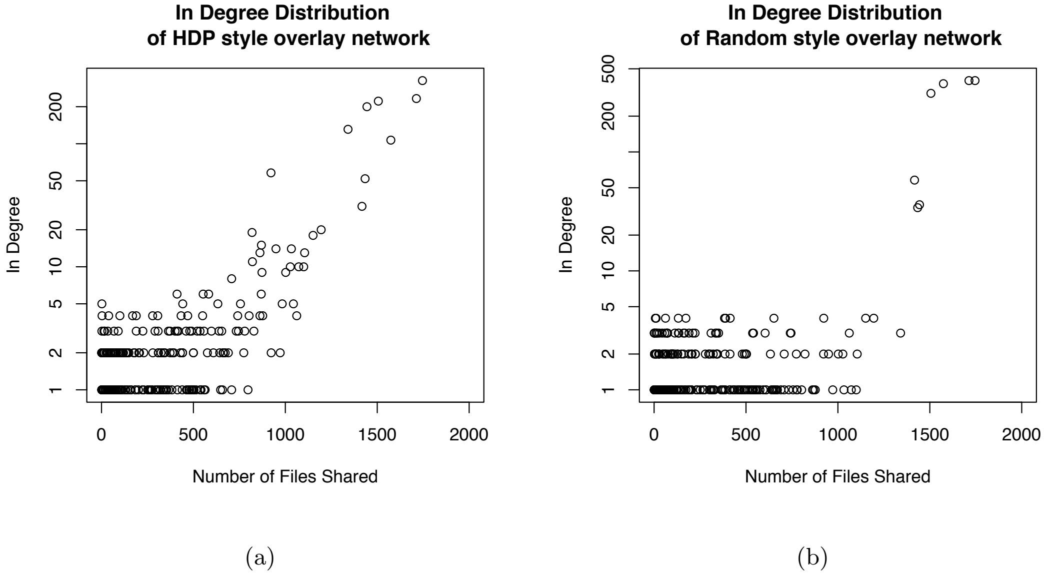
One can easily imagine collaborative filtering or a recommender system for peer-to-peer networking. Our method for creating overlay networks does not recommend songs for particular users based on the styles present in their libraries. Instead we connect users who prefer many of the same types of files. Using our overlay networks, it would be interesting to suggest possible files for download based on their presence in your neighborhood of similar users.

8. 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.

9. REFERENCES

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



(a)

(b)

Figure 9: Correlation of library size and number of connected users. Both the HDP style approach and the random style approach favor connections to users sharing many files. Due to the emphasis on user preference the HDP style approach distributes the network load among more users.

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

[3] Jacky Chu, Kevin Labonte, and Brian Neil Levine. Evaluating the use of chord with real-world peer-to-peer traces. May 2003.

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

[5] Krishna P. Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP-19)*, October 2003.

[6] Thomas 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.

[7] David Kempe, Jon Kleinberg, and Eva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of ACM SIGKDD '2003*, 2003.

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

[9] Jonathan Ledlie, Jacob M. Taylor, Laura Serban, and Margo Seltzer. Self-organization in peer-to-peer systems. In *Proceedings of European SIGOPS*, September 2002.

[10] Boon Thau Loo, Joseph M. Hellerstein, Ryan Huebsch, Scott Shenker, and Ion Stoica. Enhancing p2p file-sharing with an internet-scale query processor. In *VLDB*, 2004.

[11] Qin Lv, Sylvia Ratnasamy, and Scott Shenkar. Can heterogeneity make gnutella scalable? In *First International Workshop on Peer-to-Peer Systems (IPTPS)*, Cambridge, MA, USA, March 2002.

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