Measuring the Validity of Peer-to-Peer Data for Information Retrieval Applications

Noam Koenigstein[a], Yuval Shavitt[a], Ela Weinsberg[b], Udi Weinsberg[a]

[a] *School of Electrical Engineering, Tel-Aviv University*
[b] *Dept. of Industrial Engineering, Tel-Aviv University*

Abstract

Peer-to-Peer (p2p) networks are being increasingly adopted as an invaluable resource for various information retrieval (IR) applications, including similarity estimation, content recommendation and trend prediction. However, these networks are usually extremely large and noisy, which raises doubts regarding the ability to actually extract sufficiently accurate information.

This paper quantifies the measurement effort required to obtain and optimize the information obtained from p2p networks for the purpose of IR applications. We identify and measure inherent difficulties in collecting p2p data, namely, partial crawling, user-generated noise, sparseness, and popularity and localization of content and search queries. These aspects are quantified using music files shared in the Gnutella p2p network. We show that the power-law nature of the network makes it relatively easy to capture an accurate view of the popular content using relatively little effort. However, some applications, like trend prediction, mandate collection of the data from the "long tail", hence a much more exhaustive crawl is needed. Furthermore, we show that content and search queries are highly localized, indicating that location-crossing conclusions require a wide spread spatial crawl. Finally, we present techniques for overcoming noise originating from user generated content and for filtering non informative data, while minimizing information loss.

1. Introduction

Peer-to-Peer (p2p) networks provide a fruitful ground [1] for abundance of information, including files shared by users, search queries, and spatial and temporal changes that take place in the network. This data is often adopted as an invaluable resource for various information retrieval (IR) tasks, including user and content similarity [2, 3, 4], recommendation [5, 6], ranking [7], and trend prediction [8, 9, 10, 11].

Traditionally, these tasks use datasets extracted from server-based services, such as NetFlix, Last.FM, Yahoo! Music and other similar web 2.0 services. Web

based services have the potential to provide a complete view of their data, either by commercial agreements or by crawling using a centralized interface. P2P networks have a great potential as a practically unbounded source of data for IR tasks. This wealth of information regarding user preferences is particularly useful in recommendation techniques based on collaborative filtering, which were shown to out-perform content based approaches, given that the dataset used is sufficiently comprehensive [12].

Another advantage of p2p datasets is the availability of information, mitigating the need for agreements with website operators and various restrictions they pose on the amount of data collected or/and usage. Due to their decentralized nature and open protocols, p2p networks are a source for independent large scale data collection.

Despite all their advantages, p2p networks are quite complex, making the collection of a comprehensive dataset far from being trivial, and in some cases practically unfeasible. First, p2p networks have high user churn, as users constantly connect and disconnect from the network, being unavailable for changing periods. Second, users in p2p networks often do not expose their shared data in order to maintain high privacy and security measures, therefore disabling the ability to collect information about their shared folders. Finally, users often delete content after using it, leaving no trace of its usage.

A different complexity involves the usage of meta-data, which was shown to be useful for finding similarity between performing artists [5]. The content in file sharing networks is mostly ripped by individual users for consumption by other users. User based interactions are a desirable property in IR datasets, however when it comes to meta-data, its the main source for ambiguities and noise. Be it a movie, a song, or any other file type, typically there would be several similar duplications available on the network. The files may be digitally identical, thus having the same hash signature, yet bearing different file names, and meta-data tags. Duplication in meta-data tags are typically the result of spelling mistakes, missing data, and different variations on the correct values. A common hash signature can facilitate similar files grouping, nonetheless it does not solve the problem of copies that are not digitally identical. For example, in the Gnutella [13] network, which facilitates string-based search queries that are matched against meta-data, only 7-10% of the queries are successful in returning useful content [14].

Given the above considerations, it is clear why datasets based on p2p networks are gaining popularity in variety of IR tasks. However, the extent of data collection effort that is required in a large-scale network so that the resulting dataset would be sufficiently accurate and representative, is still unknown. The objective of this work is to bridge this gap by analyzing the efficiency and extent of crawling required in a p2p network for obtaining accurate information

for various IR tasks.

This paper quantifies the measurement effort required to obtain and optimize the information obtained from p2p networks for the purpose of IR applications. In order to understand how well the crawl captures the underlying network, we measure the utility of an exhaustive crawl relative to a partial crawl from an IR point of view. When discussing shared files, a partial crawl means that not all peers are reached, and thus not all user-to-file relations are recorded. For example, a previous study [15] reported that in Gnutella, over 30% of crawled peers are non-responsive, presumably due to firewall protections. In the context of search queries, it is practically impossible to collect all queries in a fully distributed large scale p2p network [16]. A queries based dataset is therefore destined to capture only partial view of the interactions.

Similar to previous studies [15, 17], we find that some of the graphs modeling p2p network data exhibit a power-law [18] distribution. This distribution indicates that collecting the majority of popular files and extracting accurate information for the main-streams, is relatively easy. By collecting the high degree nodes, which are easily reached, one may extract an abundance of information representing the core of user-to-content relations. On the other hand, reaching more exotic niches or following small popularity trends of digital content, mandates a more through crawl with significantly higher collection effort, as the collection process must visit the long "tail" of the distribution. Furthermore, we observe the existence of geographic locality of both files and queries, indicating that geographically aware applications like trend prediction mandate sampling in different geographic locations [10]. In additional, cultural similarities between far countries is also observed to have similar user behavior. These findings indicate a direct impact on the scalability of measurement efforts that seek to use such data in an accurate manner.

At the time of data collection, Gnutella was the most popular file sharing network [19], consisting mostly of the LimeWire client, with roughly 80-85% market share [20]. However, on October 26, 2010, US federal court ordered LimeWire to prevent "the searching, downloading, uploading, file trading and/or file distribution functionality, and/or all functionality" [21]. As a result, new LimeWire versions have been disabled, severely hurting the functionality of the network. In this research we seek to capture and interpret the behavioral patterns of users, namely the way content is shared and search queries are issued. Since content distribution properties appear similar across a range of p2p networks [22, 23], we believe that the conclusions we make can be generalized for creating accurate and optimized p2p IR applications.

2. Related Work

This paper relates to two research areas – measurement of p2p networks characteristics and the emerging fields of large-scale data mining and IR. The characteristics of various p2p networks have been extensively studied, focusing on topology, content popularity, bandwidth and queries. Most studies [24, 13, 17, 23, 25] discovered a power-law distribution in the content shared by peers in various p2p networks. Similar studies of the distribution of search queries and file replication [26, 16] in the Gnutella network also report power-law distribution.

However, Dán and Carlsson [22] recently studied the popularity of content in BitTorrent and concluded that its characteristics change depending on the measurement methodology and observation duration. The authors state that while short term or small scale measurements can conclude that the popularity of content exhibits a power-law tail, the tail in exhaustive measurements is more likely to decrease exponentially.

Since the focus of this paper is the usage of p2p networks for IR tasks, this extremely rare content that resides in the long tail, is usually less interesting, as it mostly consists of "noise" attributed to duplicated content and spelling mistakes. We empirically quantify this assumption, and show how data can be filtered in order to improve signal-to-noise ratio in a p2p based dataset.

Using p2p datasets for various IR tasks has recently began to draw attention, especially in the context of multimedia information retrieval (MIR), mostly since p2p networks hold an abundance of trendy information. This information was shown to be useful in item-based recommender systems [5, 27], collaborative filtering [4] and hype prediction [10].

However, all of the above assume that the dataset extracted from the network is sufficiently accurate. Klemm *et al.* [26] discuss the applicability of their dataset and the representativeness of their conclusions. The authors studied the fitting parameter of the power-laws observed in the query popularity distribution and replications per file for various crawl durations. The authors concluded that the relative similarity between distributions ($\gamma = 0.53 \sim 0.65$) indicate that sampling is sufficiently accurate.

This paper takes a broader look at the problem by presenting several techniques for quantifying the representativeness of p2p dataset samples for IR tasks. Since the dataset is a sample of the underlying network, it might be biased towards other features of the network such as content popularity and peer geographical location. Hence, looking solely at the distributions given different time durations is insufficient. Furthermore, we present techniques for reducing noise within the dataset, by actually leveraging the partial view of the network, hence not capturing some of the noisy long tail of the distribution.

3. Measurement Infrastructure

This section details the architecture of the measurement system developed to crawl the Gnutella [13] network and collect queries in a distributed manner. Although the exact details are adapted to comply to the Gnutella architecture and protocols, similar techniques can be applied to other p2p networks such as BitTorrent [23].

3.1. Crawling and Browsing Shared Files

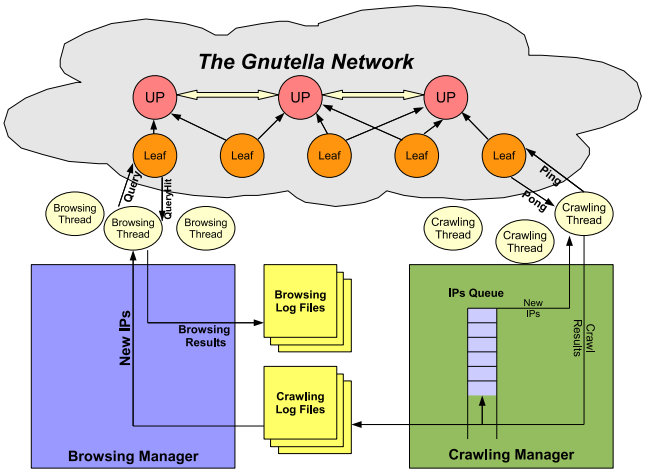


Figure 1: Crawler and Browser framework architecture

Acquiring information about shared files was achieved by developing a crawler which discovers peers and a browser which collects information about files shared by crawled peers. The framework architecture is depicted in Fig. 1. The crawler traverses the network as a graph, similar to the method used by web crawlers. The crawler employs a highly parallel technique by spawning numerous threads that attempt connecting to a set of provided IP addresses. Gnutella nodes implement a "Ping-Pong" protocol [28] used for discovering nodes, where a node that receives a "Ping" request replies with information about additional nodes that it is connected to. The resulting IP addresses are collected by the crawler, and are fed to the worker threads for further crawling.

Crawling *dynamic* p2p networks never reaches a full stop, as clients constantly connect and disconnect from the network, the crawler will always discover new IP address. This means that an "exhaustive" crawl is a matter of definition, i.e., deciding when to stop the crawling process. We use two stop conditions that define how exhaustive the crawl will be: (a) a time constraint, and (b) reaching a low rate of newly discovered nodes.

At the early stages of a crawl with an initial set of roughly 100k target node IP addresses, the rate of newly discovered nodes increases dramatically and can typically reach over 300,000 new clients per minute. As the crawling process proceeds, discovery rate slows down until it reaches a few hundreds per minute. At this point, the networks is almost fully covered, and the newly discovered nodes are mostly the ones that have joined the network only after the crawling operation started, whereas some of the crawled nodes already left the network.

The browsing operation closely follows the crawling results and operates in parallel. The browsing threads collect active node IP addresses reported from the crawler, and use a "Query" message [28] to retrieve information about the files that a node shares. Notice that some nodes ignore these queries due to privacy or bandwidth considerations.

Although we do not download any of the files, the task of browsing millions of shared folders is bandwidth intensive, and requires high bandwidth Internet access. Our deployed system uses a 1 Gbit/s network card connected to two 2.5 Gbit/s STM-16 lines. Despite our fast connection, browsing takes about 24 hours, whereas exhaustive crawling ends after less than 1 hour.

3.2. Collection of Queries

The process of query collection depends on the search paradigm that the p2p network employ. Fully distributed searches, like in Gnutella, propagate search strings between peers. While it is possible to capture a large quantity of queries by deploying several hundred "listening" nodes, it is not trivial to determine the queries origin (required for geographical location). The basic problem in identifying the origin of captured queries is that queries do not in general carry their origin IP address. Most peers are "hidden" behind a firewall, hence it is impossible to send the results directly to them. Instead, proxy peers that have routable IP address (in Gnutella – Ultrapeers) are used to convey the information for firewalled peers.

In cases where geographic query analysis is required, this usage of ultrapeers causes a difficulty to match a peer to its geographic location, since the correlation between an ultrapeer geographic location and its attached peers is low [17, 10]. The authors suggest a method to determine queries origin IP, based on the number of hops they traversed.

In the Guntella network, queries from firewalled clients do not contain the origin IP address. Instead, they contain the proxy ultrapeer's IP address. Therefore, inferring queries origin IP address is non trivial. The query collection framework was provided to us from Skyrider, a p2p advertising company, and is detailed in [17]. To understand the method, we depict a small example Gnutella network in Fig. 2.

The figure shows an intercepting node ("Skyrider Node") along with other ultrapeers and leafs (end users). Ultrapeer B is directly connected to the inter-

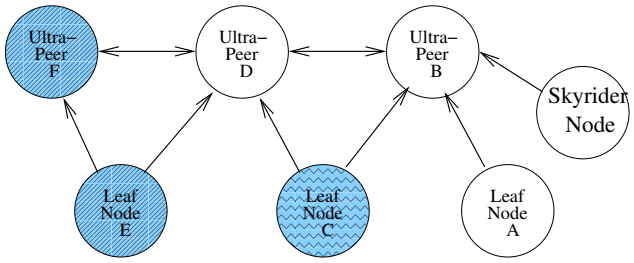


Figure 2: Query collection framework architecture

cepting node, therefore any query that traversed only a single hop must have come from it. Similarly, leaf A, leaf C (firewalled), and ultrapeer D are at a distance of two hops away, thus all queries coming from A, C and D, will have a hop count of two. Queries from Leaf A and Ultrapeer D will contain their own IP address. However, queries originating at C will contain B's IP address as C is firewalled, or otherwise unable to accept incoming connections.

Since we are directly connected to ultrapeer B, we can simply compare the query IP address with B's address. If they are not identical, the query must have come from A or D, and the address is guaranteed to be the origin address. If the query contains B's address but passed two hops, it must be coming from a firewalled client (leaf C) using ultrapeer B as a proxy. In this case, C's address is not available, and the query is not recorded. We also discard queries traveling more than 2 hops before reaching the intercepting node, as it impossible to determine their IP address. As a result, an intercepting node records traffic originating from its immediate neighborhood only (having a hop count ≤ 2), thus requiring a massive deployment of such nodes.

Other networks, e.g., BitTorrent, employ a centralized search engine, which is operated by web servers. Users search for content using a web interface, find "trackers" and use them to find active peers that hold the requested files. This technique greatly simplifies the data collection effort. However, it mandates cooperation of web site operators, which are often reluctant to share information on their users.

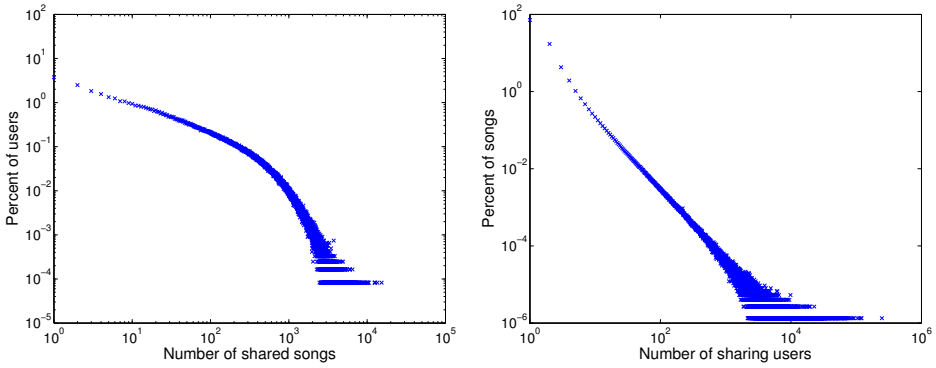
4. Dataset

The shared files dataset was collected using a 24 hours crawl of over 1.2 million users sharing 373 million files on November 25th, 2007. Most of the query analysis presented in this paper is based on a dataset collected during the first week of February 2007. It is comprised of 4.5 million queries from over 3 million users.

We identified files types according to their file suffix, and found that music related content (mp3, wma, flac, m4p, m4a) account for over 75% of the files, i.e., over 281 million files are music related. Given this, and noting that IR studies usually focus on a single domain, such as music or movies, we focus on music files in further sections.

In order to resolve queries type, we took the top-500 queries in our dataset and matched them against music records on the web. This method showed that almost 70% of the queries in the dataset are related to music, a value which serves probably as a lower bound since not all queries were identified. Although this strengthens the observation that musical content is prominent in the Gnutella network, we use all collected queries in order to avoid biased results.

5. Content Distribution



(a) Shared songs (b) Song popularity

Figure 3: Distributions of shared songs and song popularity

We begin our analysis of the dataset with understanding how content is shared by p2p users, and the way this affects the ability to create an accurate snapshot. Fig. 3a depicts distribution of songs per user, exhibiting a power-law [18] distribution, with a very strong cut-off around the middle of the plot. This distribution closely resembles the one previously reported by Zhao *et al.*[29]. The figure shows that the vast majority of users share less than 300 songs, whereas only several thousands of users share more than 1k songs. Notice that only a few users share more than 10k music files, while over 45k users share only a single song. Analysis of US-based users (56% of the users) results in a very similar power-law distribution with exactly the same exponent, making them rather representative of the entire dataset.

Next, we look at the popularity distribution of songs, by counting the number of different users that share each song. Fig. 3b shows a clear power-law distribution containing a long tail, which is attributed to popular songs that are shared by many users. This distribution was also observed in the BitTorrent p2p network, however it was argued to be partially an artifact of the locality of the sampling process [22].

The percentage of popular songs shared by many users is slightly lower in the US, yet the two distributions mostly overlap. There are a few extremely popular songs shared by more than 10k users, while the vast majority of the songs are shared by less than 1k users. Considering that there are over 1.2 million users in the dataset, songs that are shared by less than 1k users are quite borderline for being considered "popular".

The figure also shows that there are many songs that are shared by less than 100 users, which means that reaching them, or recording their relations to other songs, requires an extensive crawl. These songs surely do not represent any significant main-stream artist or genre, but for the purpose of detecting hypes or finding small communities with very specific preferences, reaching these users and collecting these songs might be important.

5.1. Filtering Noise

The above observations present different aspects of "noise", that either does not contribute insightful information or might lead towards biased analysis. Users that are "heavy sharers" might be considered as an interesting hot-spot from a networking point-of-view, however, for most IR tasks, they provide very little information since they seem to "like everything". For example, the few users that share more than 10k songs probably do not really like all the songs they share (and probably do not even know all of them). Since p2p networks do not provide us ratings, special care must be taken when considering how much the user really likes these songs, or otherwise, these users can bias the results.

Users that share only a few songs, and songs that appear at only a few users provide little or no insights for most IR tasks. For example, Fig. 3b shows that over 90% of the songs (identified using the file hash) are shared by only a single user. Again, these might be interesting network phenomenas, as they might reveal users that are "leeches" [23] or users that share illegal content [30], however, for most IR tasks, these simply contribute to the very long tail, and are hardly insightful for most IR tasks. For example, in collaborative filtering, which is presented in the following section, users that share only one song and songs that are shared by only one user, contribute no similarity relations. On the other hand, users that share songs from thousands of artists, contribute a lot of information, but some of it is likely to "pollute" the dataset with false relations, therefore the links contributed by these users are carefully handled.

A primary task which is raised from the above discussion is to correctly identify whether two given files are the same or not. For example, songs that appear only once in the dataset are mostly the result to faulty music file (not necessarily songs) or personal files that were uploaded by users, and are most probably of no interest to other users. These files are useless for most IR tasks, as they contribute no relations between users. Therefore, an accurate identification of similar files is needed in order to improve the information obtained from the network, and avoid its removal.

The simplest way to achieve this task is to use the file hash, which is a short mathematical signature of its content. Using the file hash is straightforward and suitable for all types of files, as every file in the p2p network has a file hash. However, there can be slightly different copies of the same file, each with a different hash, mostly due to small changes in the ripping process (copying the song from its media to the PC). In music content, for example, different disc ripping software or bit-rate result in different hashes of the same song.

A different form of file identification is the usage metadata, when applicable. Metadata exists for some file types, such as music and movies, and provides additional information about the file. However, this metadata is often missing and contains spelling ambiguities and mistakes, hence it can also result in incorrect identification of similar songs. These ambiguities are common in music files, for example, Lil Wayne is also spelled as "l!l w4yne". Further analysis of the genres in the dataset reveals that over 35% of the files are missing a genre, and the remaining files have over 3600 different genres.

Therefore, we use several techniques for identifying distinct songs. First, we use the file hash as the song id, and when hashes are exactly the same, we consider them as the same song. Second, accounting for the large number of songs with single appearance, only hashes that appear at least twice are included. We refer to this filter as "appearance>1". Third, we use the metadata of the song, considering only songs that have both "title" (name of the song) and "artist" tags, and use their concatenation as a unique song id. Finally, accounting for spelling mistakes and ambiguities we use SoundEx [31]. We refer to this filter as "SoundEx".

5.2. Convergence of Content

Given the above, we wish to evaluate the number of new songs that are discovered as more users are being crawled, using the different techniques for identifying distinct files.

Fig. 4 depicts the number of unique songs per number of crawled users. The order of users was randomly selected to reduce any possible spatial bias. Without using any filter, almost each crawled user adds new files. This is expected as Fig. 3b showed that over 90% of the songs are shared by a single user.

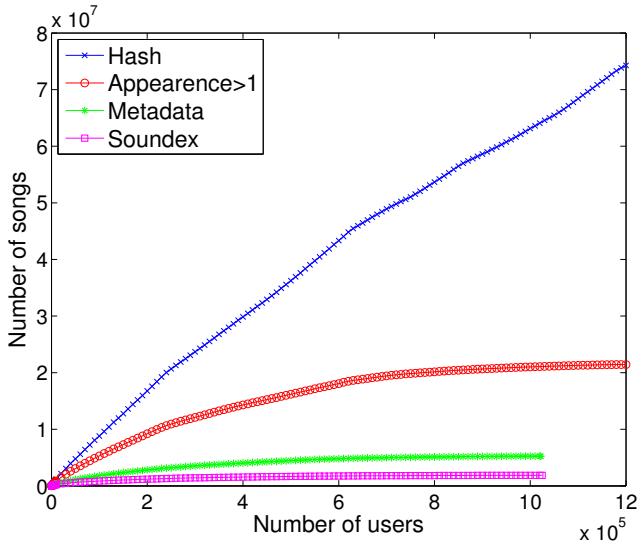


Figure 4: Convergence of songs per number of users crawled using different methods for filtering noise

However, for all noise reducing techniques, the figure shows varying convergence trends, indicating that the utility of crawling many users decreases.

The convergence witnessed when using metadata seems faster than when using file hashes, indicating that file hashes are more noisy than the metadata. Alternatively, this can be attributed to the observation that roughly 75% of the songs did not have both title and artist tags present and were therefore disregarded in that analysis. This contributes to the reduction of "noise" resulting in a more stable and quickly converging set of songs. Using SoundEx further reduces the number of shared songs and contributes to faster convergence.

Analysis of US-based users results in slower convergence than when crawling all users. The distribution of songs per US-based reveals that US-based users tend to have more songs on average, i.e., higher percentage of users have more than 200 shared songs. This explains the slower convergence, since the probability that a user will contribute previously unseen songs is higher. The number of songs seen in US-based shared folders is only half of the entire world wide collection, whereas the filtering techniques seem be as effective as in the general case, since the noise reduction remains roughly the same.

6. Collaborative Filtering

Recommendation systems [27] often require an estimation of the distance between different items [32] or between users [4]. File similarity is typically

achieved using content-based similarity [33], i.e., comparing some aspects of the content of files. This method is often computationally expensive and requires getting a hold of the actual files. Obtaining accurate user similarity is also quite challenging as it requires significant amount of information about user preferences.

However, both can be efficiently extracted from p2p networks using Collaborative Filtering (CF) methods [34], by essentially capturing the "wisdom of the crowds". The underlying assumption of the CF approach is that those who agreed in the past tend to agree again in the future. In this analysis, we use the user-to-files graph for finding similarity between files and between peers. We first transform the bi-partite user-to-files graph into a file similarity graph, where each vertex in the new graph represents a file. The weight of a link w_{ij} between two vertices i and j represents the similarity between the files, by counting the number of users that have both files in their shared folders.

6.1. Filtering Noise

We use the same song dataset as in previous sections, hence we need to carefully analyze the noise and its effect. To this end, we use the observations on the types of noise that was presented in Sec. 5.1. First, songs having hashes that appear only once in the dataset are filtered out. We then group together all file hashes that relate to identical metadata value (artist and title). At this stage we have grouped together different versions of the same song, but we yet to group different spelling variations. This is achieved by grouping together artist and title values that have a small edit distance [35] (counting insert, delete and substitute), hence accounting for small spelling mistakes. The distance threshold is dynamically determined by a function of the concatenated string length. Representative metadata values for each group are chosen using majority voting. Note that this method is finer than the previously used SoundEx, since it does not map the entire set of strings into a much shorted string. However it is significantly more computationally expensive. Finally, after this aggregation all songs that have less than 7 occurrences are removed. This value was chosen as a tradeoff between filtering and memory consumption, taking only 3bits of memory for each song.

This unification of songs reduced the number of unique songs from over 21 million identified by their hashes and 5 million when using metadata to 530k songs, which is equivalent to just 2.5% of the songs using hash and roughly 10% of the songs using metadata. Although this technique can result in over-filtering, it successfully overcomes the low signal-to-noise ratio that inherently exists in a p2p network, primarily due to user generated content. However, it is possible that some uses of p2p data would need more precise filtering techniques.

We further filter out "weak" song-to-song relations. This filter removes noise attributed by extremely "heavy sharers", and additionally, it removes malicious

and spam songs from the graph, assuming that these are not downloaded or remain in the shared folders of too many users. During the collection of songs we only include links that appear in at least 16 different users, a values which was again selected as a tradeoff between filtering and memory consumption. Finally, we keep for each file, only the top 40% links (ordered by descending similarity value) and not less than 10. After the removal of these weak links, roughly 20 million undirected links remain in the graph.

6.2. Degree Distribution

Intuitively, since some popular songs are shared by many users while many songs are shared by only a few users, it is more likely for a song to be co-shared with a popular song, hence increasing the connectivity of the popular song in the similarity graph. This type of connectivity results in a power-law degree distribution, which is characterized by a high degree of a few popular songs and a low degree of the rest. An important feature of such power-law distributions is the ability to efficiently capture many of the underlying graph properties, by sampling a partial view of the overall network [25].

On the other hand, when the "tail" of the power-law is long, i.e., many songs have very low connectivity, the crawling effort and required resources are significantly higher. The value of the data that exists in the tail greatly depends on the application [36]. Most applications do consider such "rare" files as noise, and thus their added value is marginal.

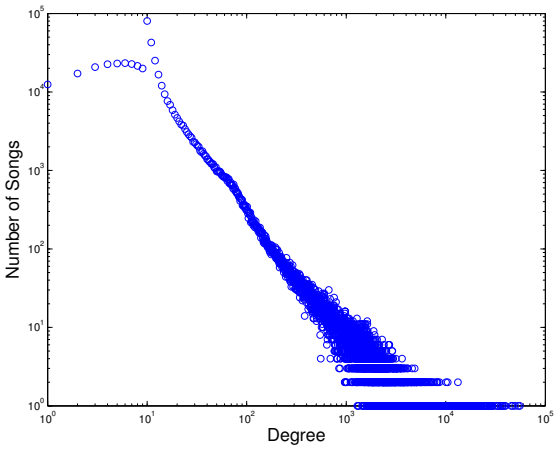


Figure 5: Degree distribution of the song similarity graph

Fig. 5 shows that the degree distribution of the similarity graph exhibits a distinct power-law behavior with a broad set of degrees. The distortion in

low degrees is attributed to the filtering of noise. By dropping songs with single appearance and setting a threshold of at least 16 on the link weight, we reduce the number of low degree songs. The second filter, that keeps at least 10 neighbors of each song increases the number of songs that have a degree of 10. This power-law degree distribution shows that even after removing many of the weaker links, there are still many songs with low connectivity. This indicates that even when not all users are reached, which is a common scenario when browsing p2p networks, there is still an abundance of information that contributes relationships between files and users.

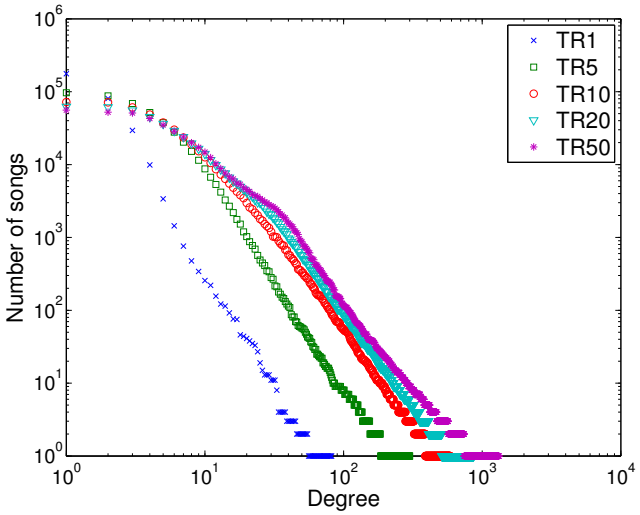


Figure 6: Song degree distribution for different sampling values of the song similarity graph

6.3. Partial Sampling

We wish to further verify that partial sampling does not significantly alter the distribution of the similarity graph. In order to account for the difference in file popularity when comparing between links, we first normalize the similarity value between any two songs. Given songs i and j, with P_i, P_j being their corresponding popularity values, w_{ij} is their non-normalized link weight, the normalized similarity is $\widehat{w}_{ij} = w_{ij}/\sqrt{P_i \cdot P_j}$.

We then create a new graph, denoted by TRn, which contains, for each file, only the top n neighbors, ordered by non increasing normalized similarity. This extends the basic filters since it uses the *normalized* similarity values, thus capturing the relative popularity of adjacent files. This filter is analogous to the effect of a partial sampling in the p2p network, where many users are simply not reached during the crawling phase. In this case, the crawl "skips" many

of the weak relations between songs, while keeping only the strong ones that appear in many users. Furthermore, it smooths the effect of the heavy sharers, that contribute links between songs that are the result of uncommon behavior, and thus are of little interest to most IR tasks. We therefore wish to evaluate the way the similarity graph is affected by partial sampling.

Fig. 6 presents the degree distribution for different values of n. The figure shows that while for n=1 the distribution is extremely sparse, reaching $n \geq 10$ results in an almost identical distribution with slightly higher node degrees. The figure also shows that the power-law exponent (indicated by the slant of the curves) is almost identical for all values of n, meaning that the connectivity pattern remains similar and mostly scales up as n increases. The effect of filtering, which is witnessed in Fig. 5, almost disappears, mostly because it smooths the effect of taking no less than 10 links for each song.

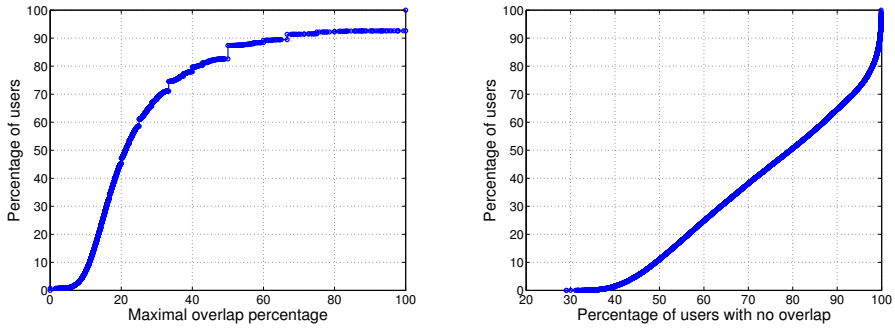
The above results indicate that obtaining partial information on the network is sufficient for generating a comprehensive and representative similarity graph, as the utility of having a more complete view of the network quickly decreases. Furthermore, heavy sharers that might bias the graph towards uncommon relationships between songs, are faded out, keeping only links that are likely to be of interest to many other users.

The above is attributed to the fact that the songs that are most affected from a partial crawl are the high-degree songs (best noticed in Fig. 6). Since many links are missed, songs that did not have too many links to begin with, are hardly affected, while songs that had many links "lose" a lot of them. Therefore, when enough links remain, i.e., a sufficient number of users that share these songs are crawled, these songs retain their high degree relative to the other songs.

6.4. Content Sparseness

CF methods often require dense dataset to operate efficiently [37]. However, as can be seen in Fig. 3a, p2p networks are often extremely sparse, mainly since each given user holds only a tiny fraction of the overall content. In fact, in our dataset only 0.03% of possible links between users and songs (identified using meta-data) actually exist, whereas in NetFlix, a social web-based movies recommendation service, which is considered very sparse, over 1% of the possible links exist. This extreme sparseness makes it difficult to assess the similarity between any two users, which is needed in order to estimate how "like-minded" they are.

In order to quantify this sparseness, we find, for each user, the maximal overlap (number of songs) it has with other users and the percentage of users it has no overlap with. Fig. 7a shows the cumulative distribution of the maximal overlap on songs between all pairs of users. The figure shows that 90% of the users have a maximal overlap of 60% with at least one more user. Moreover, 8% of the users have 100% overlap of songs with other users. However, while this



(a) Max overlap

(b) Zero overlap

Figure 7: Songs shared by a sample of 100k users, showing cumulative distributions of (a) the overlap of songs with the highest overlapping user, and (b) the percentage of users with no overlap

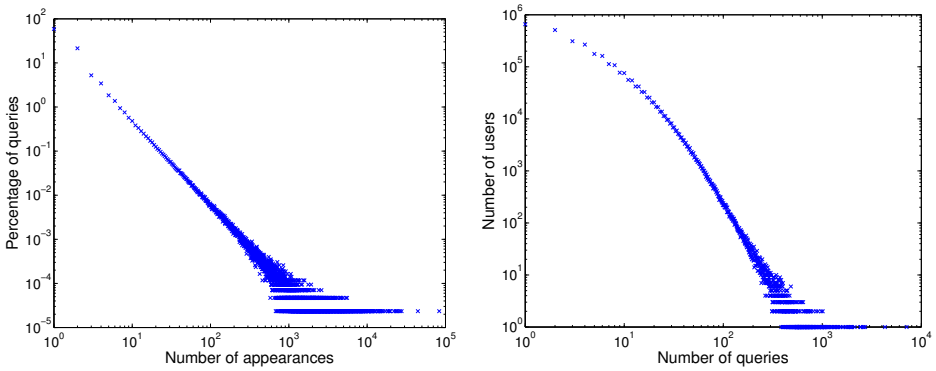
looks promising, this high overlap is mostly attributed to users with less than 5 shared songs. Furthermore, Fig. 7b shows the cumulative distribution of the percentage of users that each user has no overlap with, revealing that 50% of the users have no overlap with more than 80% of the other users, and almost 20% of the users have no overlap with any of the other users.

Such a low content overlap between users results in a "cold start" problem [38], meaning that very little information is known about a user, therefore results in difficulties for IR applications. This indicates that p2p data must be carefully processed to reduce the sparseness. This can be achieved using dimension reduction techniques (see [39] for a survey of such techniques), or otherwise by leveraging the file similarity graph [4].

7. Query Collection

Collection of queries is often a much more complicated task than crawling the shared folders [16]. Hence, similar to the previous analysis, we seek to quantify the utility of collecting queries from an increasing number of users.

We first study the distribution properties of the collected queries. Fig. 8a shows the distribution of query appearances, depicting a distinct power-law distribution, similar to previous work [40]. This indicates that only a few popular search terms repeat at a high frequency, whereas roughly 65% of the queries appear only once. Fig. 8b shows the "popularity" of queries, i.e. the distribution of queries per users. The figure shows that a significant portion of users issued less than 10 queries during the entire week of sampling. These distributions can be attributed again to noise in the p2p dataset, which should be reduces.



(a) Query distribution (b) Query popularity

Figure 8: Unique query distribution and convergence per number of crawled users

7.1. Filtering Noise

Noise in queries is mostly related to either scarce keywords, or more likely, spelling mistakes. Therefore, we use techniques similar to the above: (a) keep only queries that appear more than once, and (b) aggregate queries using SoundEx algorithm.

7.2. Convergence of Queries

Fig. 9 depicts the number of unique queries per number of crawled users. The figure shows that when all the queries are considered, the convergence is very slow (note the semi-log scale), meaning that each additional user contributes some new queries. When considering only queries that appeared more than once, there is a more noticed convergence, and the overall number of unique queries goes down to less than 2 million. Applying SoundEx on the data significantly removes the number of distinct queries and results in an amazingly fast convergence. As expected, the graphs shows that spelling mistakes are prominent in queries, in particular, much more than in metadata.

This analysis shows that the diversity in search terms is mostly attributed to very "rare" strings that originate from single users, and to spelling mistakes, whereas the majority of the common queries are frequently repeating amongst the different users, hence can be more easily collected. Filtering one time queries is justified because such queries account for many of the spelling mistakes in search terms, and useless for collaborative filtering. Applying the more intense SoundEx filter almost completely mitigates duplications due to spelling mistakes, but probably groups together some different queries as well.

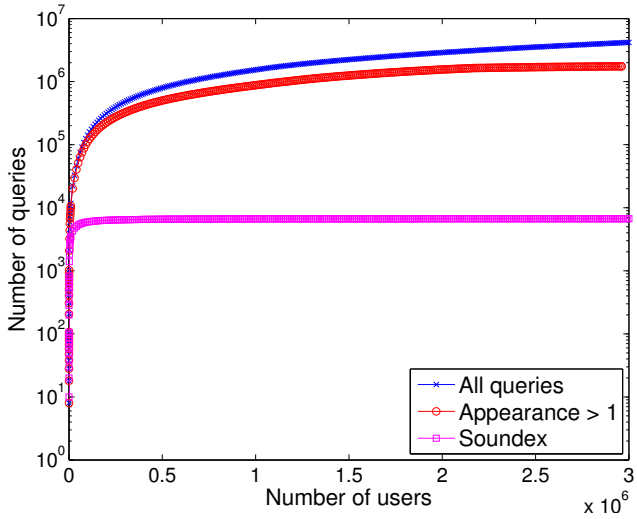


Figure 9: Convergence of queries per number of crawled users using different methods for filtering noise

7.3. Spatial Correlation

Queries were shown to be highly correlated with geographic location [16, 17]. In order to quantify the implications of localized query collection, we compared the top-1000 string queries performed by peers in the US and different countries. We define the correlation as the total number of matching strings (max value is 1000).

Fig. 10 depicts the number of matching queries between the US and several other countries over a period of 17 weeks in early 2007. As expected, the figure shows high correlation between English speaking countries (Australia and the UK) and the US, whereas the non-English speaking countries exhibit much lower similarity. Japan appears to have the lowest overall correlation with US, with less than 20 matching queries out of 1000.

Interestingly, the correlation is quite consistent over the entire period, showing a profound difference between the Anglosphere countries and other countries. Putting aside the musical anthropology aspects of these results, this analysis indicates that when performing targeted research, it is sufficient to focus on a bounded geographical region or country. However, conclusions drawn using queries collected in a specific region should be carefully examined before inferred in other geographical locations.

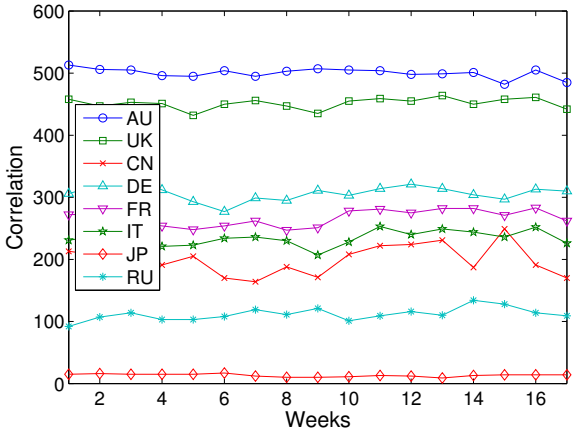


Figure 10: Number of matching countries between top-1000 search queries in the US and different countries over extending period

8. Discussion and Conclusion

In face of the increasing usage of p2p networks in IR tasks, this paper provides a comprehensive analysis of different aspects that must be considered. Four main challenges are described: (a) the inability to crawl all users and collect information regarding all files, (b) the complexities in intercepting search queries, (c) the inherent noise of user generated content, and (d) the extreme sparseness of the dataset.

Content in p2p networks typically follows a power-law distribution, hence discovering the popular relations is relatively easy. Partial crawling is shown to have much less impact on the availability of popular content than on specific "niches". On the other hand, when popularity is considered, partial sampling is more likely to effect the popular files. Although their relative popularity decreases, file-to-file relations remain intact. Additionally, since the network is extremely sparse, partial crawling does not significantly change the observation that proper handling of this data is required, even in the presence of an exhaustive crawl.

Spatial analysis reveals that p2p networks are highly localized, with profound differences in files and queries between geographical regions. This can help inducing a localized research regarding musical trends and preferences, but mandates careful consideration before inferring conclusion drawn from local samples.

File sharing networks have low signal-to-noise ratio, mandating careful data processing when compared to "traditional" datasets (e.g., website). In order to improve the ability to extract insightful information from the data, we suggest

removing files that appear only once in the dataset, and peers that share too many songs, meaning, removing the extremes that are not insightful and may "pollute" the dataset. Furthermore, we present methods for file identification that help merge similar files, further improving the signal-to-noise ratio. This extensive filtering can be applied to reduce redundant records and false relations, but may result in loss of data, which can be of interest to other IR tasks, such as popularity predictions.

Overall, p2p networks provide an abundance of information that can be utilized in IR research. Main-stream data can be easily collected from p2p networks, while having all the benefits over standard website data. However, when seeking to harness the power of the long tail, where p2p networks have a significant advantage, careful analysis is essential for sufficient noise reduction while maintaining relevant information.

References

[1] Y. Shavitt, E. Weinsberg, U. Weinsberg, Mining musical content from large-scale peer-to-peer networks, in: Multimedia Magazine, IEEE, 2011.

[2] D. P. W. Ellis, B. Whitman, The quest for ground truth in musical artist similarity, in: The International Society for Music Information Retrieval (ISMIR), 2002.

[3] A. Berenzweig, B. Logan, D. P. W. Ellis, B. Whitman, A large-scale evaluation of acoustic and subjective music similarity measures, in: Computer Music Journal, 2003.

[4] Y. Shavitt, E. Weinsberg, U. Weinsberg, Estimating peer similarity using distance of shared files, in: International Workshop on Advances in Music Information Research (adMIRe), 2010.

[5] Y. Shavitt, U. Weinsberg, Song clustering using peer-to-peer co-occurrences, in: International workshop on Peer-To-Peer Systems (IPTPS), 2009.

[6] I. Mierswa, K. Morik, M. Wurst, Collaborative use of features in a distributed system for the organization of music collections, in: Intelligent Music Information Systems: Tools and Methodologies, Idea Group Publishing, 2007, pp. 147–176.

[7] N. Koenigstein, Y. Shavitt, Song ranking based on piracy in peer-to-peer networks, in: The International Society for Music Information Retrieval (ISMIR), 2009.

[8] S. Bhattacharjee, R. D. Gopal, K. Lertwachara, J. R. Marsden, Using P2P sharing activity to improve business decision making: proof of concept for estimating product life-cycle, Electronic Commerce Research and Applications 4 (1) (2005) 14–20.

[9] S. Bhattacharjee, R. Gopal, K. Lertwachara, J. R. Marsden, Whatever happened to payola? an empirical analysis of online music sharing, Decis. Support Syst. 42 (1) (2006) 104–120.

[10] N. Koenigstein, Y. Shavitt, T. Tankel, Spotting out emerging artists using geo-aware analysis of p2p query strings, in: ACM KDD '08, 2008, pp. 937–945.

[11] N. Koenigstein, Y. Shavitt, Predicting billboard success using data-mining in p2p networks, in: International workshop on Peer-To-Peer Systems (IPTPS), 2009.

[12] L. Barrington, R. Oda, G. Lanckriet, Smarter than genius? human evaluation of music recommender systems, in: The International Society for Music Information Retrieval (ISMIR), 2009.

[13] M. Ripeanu, Peer-to-peer architecture case study: Gnutella network (2001).

[14] M. A. Zaharia, A. Chandel, S. Saroiu, S. Keshav, Finding content in file-sharing networks when you can't even spell, in: International Workshop on Advances in Music Information Research (adMIRe), 2007.

[15] D. Stutzbach, R. Rejaie, Characterizing the two-tier gnutella topology, in: SIGMETRICS, ACM, 2005.

[16] A. Klemm, C. Lindemann, M. K. Vernon, O. P. Waldhorst, Characterizing the query behavior in peer-to-peer file sharing systems, in: IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, ACM, New York, NY, USA, 2004.

[17] A. S. Gish, Y. Shavitt, T. Tankel, Geographical statistics and characteristics of p2p query strings, in: International Workshop on Advances in Music Information Research (adMIRe), 2007.

[18] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, SCIENCE 286 (1999) 509 – 512.

[19] P. Resnikoff, Digital media desktop report: Fourth quarter of 2007, digital Music Research Group.

[20] A. Rasti, D. Stutzbach, R. Rejaie, On the long-term evolution of the two-tier Gnutella overlay, in: Global Internet, Barcelona, Spain, 2006.

[21] Wikipedia: Liemwire, http://en.wikipedia.org/wiki/LimeWire.

[22] G. Dán, N. Carlsson, Power-law revisited: A large scale measurement study of p2p content popularity, in: International Workshop on Advances in Music Information Research (adMIRe), 2010.

[23] G. Siganos, J. Pujol, P. Rodriguez, Monitoring the Bittorrent monitors: A bird's eye view, in: PAM, 2009.

[24] M. Ripeanu, I. Foster, A. Iamnitchi, Mapping the Gnutella network: Properties of large-scale peer-to-peer systems and implications for system design, IEEE Internet Computing Journal 6 (2002) 2002.

[25] M. Faloutsos, P. Faloutsos, C. Faloutsos, On power-law relationships of the Internet topology, in: SIGCOMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication, 1999, pp. 251–262.

[26] A. Klemm, C. Lindemann, O. P. Waldhorst, Relating query popularity and file replication in the gnutella peer-to-peer network, in: GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems, 2004.

[27] P. Resnick, H. R. Varian, Recommender systems, Communications of the ACM 40 (3).

[28] The Gnutella protocol specification v0.41, http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf (2010).

[29] S. Zhao, D. Stutzbach, R. Rejaie, Characterizing files in the modern Gnutella network: A measurement study, in: SPIE/ACM Multimedia Computing and Networking, 2006.

[30] M. Latapy, C. Magnien, R. Fournier, Quantifying paedophile queries in a large P2P system, in: INFOCOM, 2011, pp. 401–405.

[31] D. Knuth, The Art of Computer Programming, Vol. 3: Sorting and Searching, Addison-Wesley, 1973.

[32] O. Celma, Music recommendation and discovery in the long tail, Ph.D. thesis, Universitat Pompeu Fabra, Barcelona (2008).

[33] M. S. Lew, Content-based multimedia information retrieval: State of the art and challenges, ACM Trans. Multimedia Comput. Commun. Appl 2 (2006) 1–19.

[34] J. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, in: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1998, pp. 43–52.

[35] R. A. Wagner, M. J. Fischer, The string-to-string correction problem, J. ACM 21 (1) (1974) 168–173.

[36] O. Celma, P. Cano, From hits to niches? or how popular artists can bias music recommendation and discovery, in: 2nd Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition, Las Vegas, USA, 2008.

[37] J. L. Herlocker, J. A. Konstan, L. G. Terveen, John, T. Riedl, Evaluating collaborative filtering recommender systems, ACM Transactions on Information Systems 22 (2004) 5–53.

[38] A. I. Schein, A. Popescul, L. H. Ungar, D. M. Pennock, Methods and metrics for cold-start recommendations, in: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2002, pp. 253–260.

[39] I. Fodor, A survey of dimension reduction techniques, Tech. rep., Center for Applied Scientific Computing, Lawrence Livermore National Laboratory (2002).

[40] K. Sripanidkulchai, The popularity of gnutella queries and its implications on scalability (2001).