ON THE APPLICABILITY OF PEER-TO-PEER DATA IN MUSIC INFORMATION RETRIEVAL RESEARCH

Noam Koenigstein[1], Yuval Shavitt[1], Ela Weinsberg[2], and Udi Weinsberg[1]

[1]School of Electrical Engineering, Tel-Aviv University
[2]Dept. of Industrial Engineering, Tel-Aviv University

ABSTRACT

Peer-to-Peer (p2p) networks are being increasingly adopted as an invaluable resource for various music information retrieval (MIR) tasks, including music similarity, 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 evaluates the applicability of using data originating from p2p networks for MIR research, focusing on partial crawling, inherent noise and localization of songs and search queries. These aspects are quantified using songs collected from 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 main-streams 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 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 are being increasingly adopted as an invaluable resource for various music information retrieval (MIR) tasks [11], including music and user similarity [3, 5, 15], recommendation [16], ranking [9, 14], and even trend prediction [10, 12]. Various information can be extracted from a p2p network, including files shared by users, search queries, and spatial and temporal changes that take place in the network.

This type of information is traditionally extracted from server-based services, such as Last.FM and Yahoo! Music 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. However, while p2p networks have practically unbounded growth potential, web-based services are often limited in size. This limitation is problematic for collaborative filtering techniques, that were shown to out-perform content based approaches, given that the dataset used is sufficiently comprehensive [2].

Another advantage of p2p datasets over traditional datasets is the availability of information, mitigating the need for agreements with website operators and various restrictions they pose on the data 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, causing users to 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 shared files to save space making it invisible to a crawl being performed after the deletion.

It is yet unknown to what extent data that is collected from large-scale p2p networks actually represents sufficiently accurate information in general, and particularly from a MIR point of view. The objective of this work is to bridge this gap by analyzing the efficiency and extent of crawling required for obtaining accurate information for various MIR tasks. We focus on sufficient sampling in a sparse domain with a long tail of content distribution.

In order to understand how well the crawl captures the underlying network, we perform an empirical study of the utility of an exhaustive crawl relative to a partial crawl. When discussing shared files, a partial crawl means that not all users are reached, resulting in not all songs being collected. Additionally, in the context of search queries, only a portion of the queries are collected since it is practically impossible to collect all queries in a fully distributed p2p network.

We find that some of the graphs modeling p2p network data exhibit a power-law [1] 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 regarding the core of the network. On the other hand, reaching more exotic niches or following small changes in trendy hits 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 applications that are geographic aware (like trend prediction [10]), mandate sampling from different geographic locations.

2. 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. As such is Apollo [17], an efficient framework for crawling the BitTorrent p2p network, which uses a centralized server that collects trackers, enabling it to reach related peers and extract files that peer hold.

2.1 Crawling and Browsing Shared Files

Our 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 [18] 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 fed to the worker threads for further crawling.

Crawling *dynamic* p2p networks never reaches a complete stop, as clients constantly connect and disconnect from the network, and the crawler keeps discovering 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 network 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 [18] 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 crawling ends after roughly 1 hour. More details on our crawler can be found in [8].

2.2 Collection of Queries

The process of query collection is highly dependant on the search paradigm that the p2p protocol employs. 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 [7, 10]. The authors suggest a method to determine queries origin IP, based on the number of hops they traversed. Our geographical resolution is based on a similar technique. More details can be found in [7].

Alternatively, some 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.

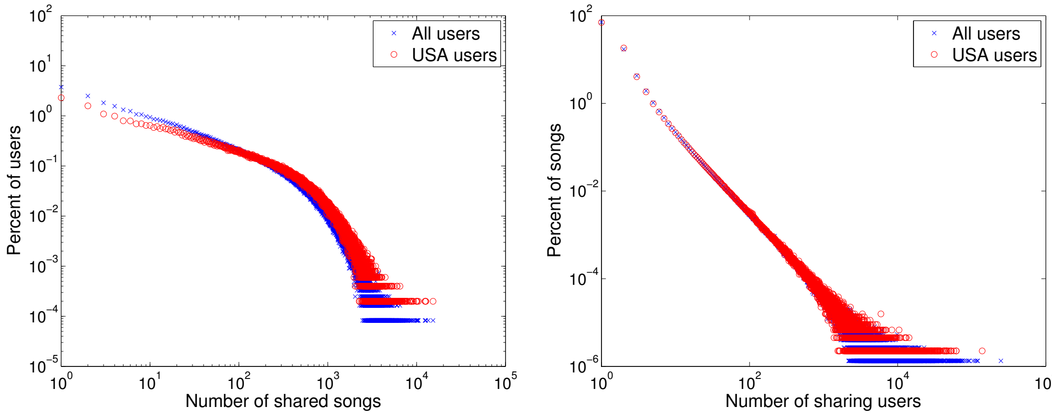
3. SONG DISTRIBUTION

We start by looking at the distribution of songs per users, considering all users in the dataset, and only users that are located in the US. For this end, we consider only music files shared by users, namely files ending with .mp3, .wav, .mid and .wma.

Figure 1(a) shows that all users and US-only users exhibit a power-law [1] distribution, with a very strong cutoff around the middle of the plot. This indicates 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.

These two extremes present different aspects of "noise". The few "heavy sharers" are not informative, while the latter simply contribute to a very long tail that is hardly insightful. In collaborative filtering for example, users that share only one song, contribute no similarity relations, while users that share songs from thousands of artists, are likely to "pollute" the dataset with false relations, since they appear to "like everything".

Next, we look a the popularity distribution of songs, by counting the number of different users that share each song. Figure 1(b) shows a clear power-law distribution containing a long tail, which is attributed to popular songs that are shared by many users. 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,



(a) Shared songs

(b) Song popularity

Figure 1. Distributions of shared songs and song popularity

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.

Given these distributions, we wish to evaluate the number of new songs that are discovered as more users are being crawled. Two difficulties arise regarding this analysis. The first is the way to identify that two files are indeed the same song, and for this end, either the file hash or the metadata can be utilized. Using the file hash is straightforward, as every file in the p2p network has a file hash, taken over its content. However, there can be many slightly different copies of the same file, each with a different hash, mostly due to different disc ripping software or originating song. On the other hand, metadata is often missing and contains different spelling mistakes, hence it can also result in incorrect identification of similar songs.

Therefore, we used both file hash and metadata techniques for identification of unique songs. First, we just use the file hash as the song id, and when hashes are exactly the same, we consider them as the same song. When using metadata, we consider only songs that have both “title” (name of the song) and “artist” tags, and use their concatenation as the song id.

The second difficulty is that many songs appear only once in the dataset. These are mostly attributed to faulty music file (not necessarily songs) that were uploaded by users and are of no interest to other users, rendering these files is useless for most MIR tasks. Therefore, we first counted the number of occurrences of each song, once using file hash and then using metadata, and removed all the songs that have only a single appearance in the dataset.

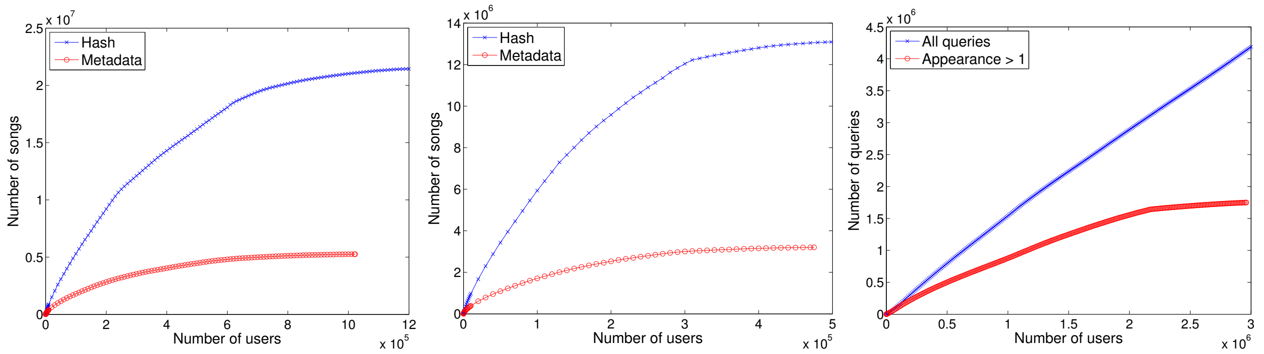
Figure 2 shows the number of unique songs per number of crawled users, showing all users and US-based users. The order of users was randomly selected to reduce spatial bias. Both figures show a converging trend, indicating that the utility of crawling many users decreases. Furthermore, the convergence 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, hence were removed from the analysis. This contributes to the reduction of “noise” resulting in a more stable and quickly converging set of songs.

The convergence observed when crawling only US-based users (56% of the users) seems slower than when crawling all users. Looking back at the distribution of songs per users (Figure 1(a)) shows that US users tend to have more songs, 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. However the usage of metadata over hash for songs identification seem to be as effective as in the general case, since the percentage of noise reduction remains the same.

4. SONG CONNECTIVITY

Item-based recommendation systems require an estimation of the distance between songs. This task is often performed using expensive content-based similarity. However, song similarity can be efficiently extracted from p2p networks, by transforming the bipartite graph that connects users to songs into a 1-mode song-similarity graph, where the weight of a link w_{ij} between two songs i and j is the number of users that have both songs in their shared folders.

In this analysis we wish to obtain a stable similarity graph, therefore we do more processing to identify unique songs. Similar to the previous analysis, all the songs that have hash value that appeared only once are removed. We then group together all file hashes that relate to identical metadata value (artist and title). At this stage we have grouped together different digital versions of the same song. Accounting for spelling mistakes is achieved by grouping together artist and title values that have a small edit distance [19] (counting insert, delete and substitute). The dis-



(a) All users

(b) US-based users

(c) Unique queries

Figure 2. Number of unique songs (using file hash and metadata) and unique queries vs. number of users crawled

tance threshold is determined by a function of the string's length. Representative metadata values are chosen using majority voting. Finally, after this aggregation, all songs that have less than 7 occurrences are removed. This value is 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 when using hashes and 5 million when using metadata to 530k songs, meaning only 2.5% of the songs using hash and roughly 10% of the songs using metadata. Although this technique can slightly overfilter, it successfully overcomes the low signal-to-noise ratio that inherently exists in the p2p network, primarily due to user generated content.

We further perform filtering of "weak" song-to-song relations, to remove noise as the one witnessed in the presence of extremely "heavy sharers". 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. Then, we kept for each file, only the top 40% links (ordered by descending similarity value) and not less than 10. Notice that this filter also removes malicious and spam songs from the graph, assuming that these are not downloaded by too many users. After the removal of these "weak" links, roughly 20 million undirected links remain in the graph.

4.1 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 results in high degrees of the few popular songs and lower degree of many less-popular songs. 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.

On the other hand, when the "tail" of the power-law is long, meaning 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 [4]. Most applications do consider such "rare" files as noise; in that case, their added value is marginal.

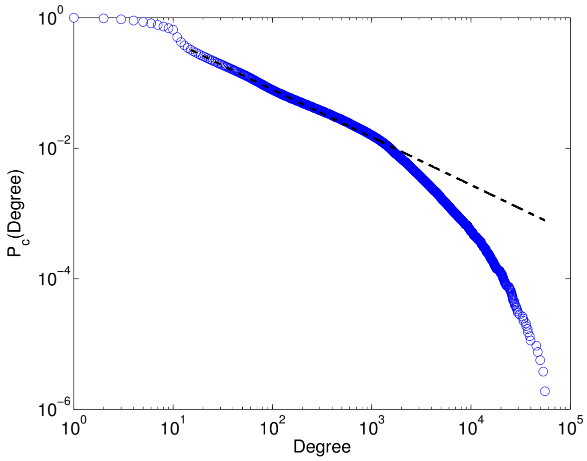


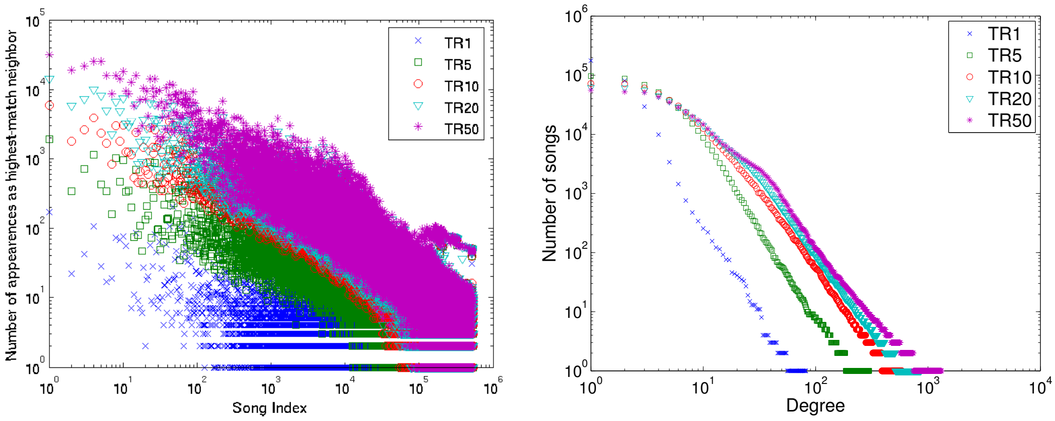
Figure 3. Cumulative degree distribution of the song similarity graph

Several previous studies on p2p networks [6, 7] show that graphs that model various p2p networks exhibit power-law distributions. As can be seen in Figure 4.1 shows the cumulative song degree distribution in the similarity graph, exhibiting a power-law with a strong cut-off. This power-law distribution suggests that there are relatively a few songs with very high connectivity and many songs with low connectivity.

4.2 Partial Sampling

We wish to verify that partial sampling does not significantly alter the distribution of the similarity graph. We first normalize the similarity value between any two songs so it reflects their popularity. Hence, the new similarity is $\widehat{w}_{ij} = w_{ij}/\sqrt{P_i \cdot P_j}$, where w_{ij} is the link weight between songs i and j, and P_i, P_j are their corresponding overall number of occurrences (popularity).

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



(a) Top relations

(b) Degree distribution

Figure 4. Effect of sampling on song similarity distribution

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. We therefore, wish to evaluate the way the similarity graph is affected by partial sampling.

The number of times each song appears as the nearest neighbor for different values of N is presented in Figure 4(a). The figure shows that for N=1,5 the distributions are significantly different, whereas for $N \geq 10$ the distributions almost overlap. Similar results can be seen when looking at the degree distribution depicted in Figure 4(b). 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 above results indicate that obtaining partial information on the network is sufficient for generating a comprehensive similarity graph, as the utility of having a more complete view of the network quickly decreases. This is attributed to the fact that the songs that are most affected from this partial crawl are the high-degree songs (best noticed in Figure 4(b)). Since many links are gone, 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. However, when enough links remain (a sufficient number of users that share these songs are crawled), these songs retain their high degree relative to the other songs.

5. QUERY COLLECTION

Collection of queries is often a much more complicated task than crawling the shared folders. Hence, we seek to quantify the utility of collecting queries from an increasing number of users, similar to the way we did for unique songs. For this end, we collected almost 4.5 million queries from over 3 million users during a week in February 2007. Notice that these queries are not related only to music, however analysis of keywords used for searching the Gnutella network shows that almost 70% of the queries are music related [10].

Figure 2(c) depicts the number of unique queries per number of crawled users, using all the queries, and using only queries that appeared more than once. The figure shows that when all the queries are considered, there is no convergence, meaning that each additional user contributes some new queries. However, when we consider only queries that appeared more than once, there is a clear convergence, and the overall number of unique queries goes down to less than 2 million. We therefore, learn that the diversity in search terms is mostly attributed to very “rare” strings that originate from single users, whereas the majority of the common queries are frequently repeating amongst the different users, hence can be more easily reached.

Queries were shown to be highly correlated with geographic location [7], which is rather intuitive considering the cultural and language differences between countries. In order to quantify the implications of localized query collection, we compared the top-1000 string queries performed by users in different countries, and define the correlation as the total number of matching strings.

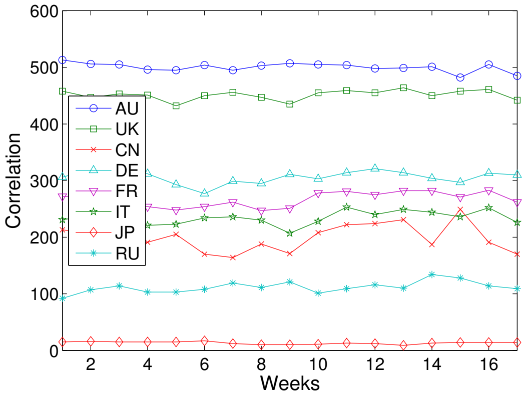


Figure 5. Correlation between top-1000 search queries between the US and different countries over time

Figure 5 depicts the correlation factor between the US and other countries over a period of 17 weeks in early 2007. The figure shows that, as expected, the English speaking countries (Australia and United Kingdom) have much higher correlation with the US than the non-English speaking countries. Japan appears to have the lowest overall correlation, with less than 20 matching queries. Interestingly, the correlation is quite consistent over the entire period, showing profound differences between the Anglosphere and the non-English speaking 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 assuming them on other geographical locations.

6. DISCUSSION AND CONCLUSION

In the presence of an increasing demand for large scale datasets in MIR research, this paper investigates the different considerations in using a p2p based dataset. Several difficulties are considered – the inability to crawl all users and collect information on all songs, the complexities in intercepting all search queries and the inherent noise of user generated content.

Content distribution in a p2p networks typically exhibits a power-law, hence collecting the majority of songs is rather easy. Partial crawling is shown to have much less impact on the availability of main-stream content than on specific "niches". On the other hand, when popularity is considered, partial sampling is more likely to effect the popular songs. Although their relative popularity decreases, song-to-song relations remain intact.

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

File sharing networks were shown to 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 songs that appear only once in the dataset, and users that share too many songs, therefore, removing the extremes that are not insightful and may "pollute" the dataset. Furthermore, we present methods for song identification that help merge similar songs, 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 some MIR tasks, such as popularity predictions.

Overall, p2p networks provide an abundance of information that can be utilized in MIR 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 key for sufficient noise reduction while maintaining relevant information.

Acknowledgment. This research was supported in part by a grant from the Israel Science Foundation (ISF) center of excellence program (grant number 1685/07).

7. REFERENCES

[1] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *SCIENCE*, 286:509 – 512, 1999.

[2] Luke Barrington, Reid Oda, and Gert Lanckriet. Smarter than genius? human evaluation of music recommender systems. In *ISMIR*, 2009.

[3] Adam Berenzweig, Beth Logan, Daniel P. W. Ellis, and Brian Whitman. A large-scale evaluation of acoustic and subjective music similarity measures. In *Computer Music Journal*, 2003.

[4] Oscar Celma and Pedro Cano. From hits to niches? or how popular artists can bias music recommendation and discovery. In *2nd Workshop on Large-Scale Recommender Systems*, 2008.

[5] Daniel P. W. Ellis and Brian Whitman. The quest for ground truth in musical artist similarity. In *ISMIR*, 2002.

[6] F. Le Fessant, A. M. Kermarrec, and L. Massoulie. Clustering in peer-to-peer file sharing workloads. In *IPTPS*, 2004.

[7] Adam Shaked Gish, Yuval Shavitt, and Tomer Tankel. Geographical statistics and characteristics of p2p query strings. In *IPTPS*, 2007.

[8] Noam Koenigstein, Gert Lanckriet, Brian McFee, and Yuval Shavitt. Collaborative filtering based on p2p networks. In *ISMIR*, Utrecht, the Netherlands, 2010.

[9] Noam Koenigstein and Yuval Shavitt. Song ranking based on piracy in peer-to-peer networks. In *ISMIR*, 2009.

[10] Noam Koenigstein, Yuval Shavitt, and Tomer Tankel. Spotting out emerging artists using geo-aware analysis of p2p query strings. In *KDD*, 2008.

[11] Noam Koenigstein, Yuval Shavitt, Ela Weinsberg, Udi Weinsberg, and Tomer Tankel. A framework for extracting musical similarities from peer-to-peer networks. In *AdMIRe*, Singapore, July 2010.

[12] Noam Koenigstein, Yuval Shavitt, and Noa Zilberman. Predicting billboard success using data-mining in p2p networks. In *AdMIRe*, 2009.

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

[14] Markus Schedl, Tim Pohle, Noam Koenigstein, and Peter Knees. What's Hot? Estimating Country-Specific Artist Popularity. In *ISMIR*, Utrecht, the Netherlands, August 2010.

[15] Yuval Shavitt, Ela Weinsberg, and Udi Weinsberg. Estimating peer similarity using distance of shared files. In *IPTPS*, 2010.

[16] Yuval Shavitt and Udi Weinsberg. Song clustering using peer-to-peer co-occurrences. In *AdMIRe*, 2009.

[17] Georgos Siganos, Josep Pujol, and Pablo Rodriguez. Monitoring the Bittorrent monitors: A bird's eye view. In *PAM*, 2009.

[18] The Gnutella Protocol Specification v0.41. `http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf`, 2010.

[19] Robert A. Wagner and Michael J. Fischer. The string-to-string correction problem. *J. ACM*, 21(1):168–173, 1974.