A FRAMEWORK FOR EXTRACTING MUSICAL SIMILARITIES FROM PEER-TO-PEER NETWORKS

Noam Koenigstein Yuval Shavitt Tomer Tankel Ela Weinsberg Udi Weinsberg

School of Electrical Engineering
Tel Aviv University

ABSTRACT

The usage of peer-to-peer (p2p) networks for music information retrieval (MIR) tasks is gaining momentum. P2P file sharing networks can be used for collecting both search queries and files from shared folders. The first can be utilized to reveal current taste, users interest, and trends, while the latter can be used for enhancing recommender systems. Both provide opportunities for longitudinal analysis, as queries change over time and content often accumulates. Moreover, spatial analysis can expose cultural differences and the way trends propagate. However, tapping into this fountain of information is far from trivial.

This paper presents a novel analysis of the shared folders data-set collected from the Gnutella network. We first present the framework for crawling the network and collecting the data. We then present some data-set characteristics, while focusing on music similarities. The paper sheds light on both the opportunities of using p2p data and its complexities.

Keywords— File-sharing, Peer-to-peer, Information Retrieval, Data-mining

1. INTRODUCTION

Peer-to-Peer (p2p) networks are gaining momentum in a variety of music information retrieval (MIR) tasks, ranging from similarity measures [1, 2], recommendation systems [3], trend prediction [4, 5], and artists ranking [6]. Despite the controversy over legal issues, the number of users and available content is in on the rise. P2P networks are therefore, an excellent source for learning the relations between users and their favorite music. It is important to note that collecting this information does not require actual sharing or downloading any illegal content.

MIR research that is based on statistical interpretations such as Collaborative Filtering (CF), which essentially capture the "wisdom of the crowds", often perform better than content based approaches [7, 8]. This gap might be attributed to the disregard of cognitive information that is not in the signal [9].

Human studies have shown that music recommendation based on collaborative filtering outperform content-based approaches so long as the data-set used is sufficiently comprehensive [10]. However, when the data-set is insufficient, or the artists are less popular (those in the long tail), content-based approaches have an advantage. The scale of a CF data-set and the diversity of users are therefore of great importance.

Data collection in file sharing networks typically reach very large scales. A 24 hours crawl of the Gnutella network may result in over 200 million user-to-song relation from over a million users. A scale that is much larger than "traditional" data-sets which typically originate from social web services. For example, the well established Last.FM data-set provided by [11], consist of 17.6 million user-to-song relations from almost 360k users.

A second advantage of p2p data-sets over traditional data-sets is the availability of information. Content based data-sets mandate access to the actual songs, making them both computationally intense and costly. In social networks or websites such as Last.FM, data collection is dependent on the goodwill of the website owner, and restrictions are often set on the amount of information that can be collected. Other large-scale data-sets are the property of commercial companies (Google, Apple, Yahoo! etc.), which are usually reluctant to share it. Due to their decentralized nature and open protocols, p2p networks are a source for independent large scale data collection. Anyone who overcomes the initial technological barrier can crawl the network and collect valuable information.

P2P networks are a fertile ground for an abundance of MIR related information. Media files, such as MP3 files, include ID3 tags that reveal information such as the title, artist, album and track number. Although these records are sometimes absent or conflicting, it is still possible to restore some of the correct values. Moreover, p2p data-sets typically include the IP addresses of the users. This allows for accurate geographical positioning of users. Such information is valuable for a variety of tasks, such as user classification, community detection [12] and trend prediction [4].

This paper presents a framework for collecting and analyzing music data from the Gnutella network. We perform a statistical analysis of our shared folders data-set. We focus our discussion on user similarity and song similarity, which are commonly used in CF systems and item-based recommender systems. The entire p2p data-set used in this paper will be available as a contribution to the MIR community on the authors website by publication time.

The remainder of this paper is organized as follows. Section 2 details the framework for crawling the network and collecting the data. We then analyze a snapshot of the collected data, and present its statistics and usage for inferring similari-

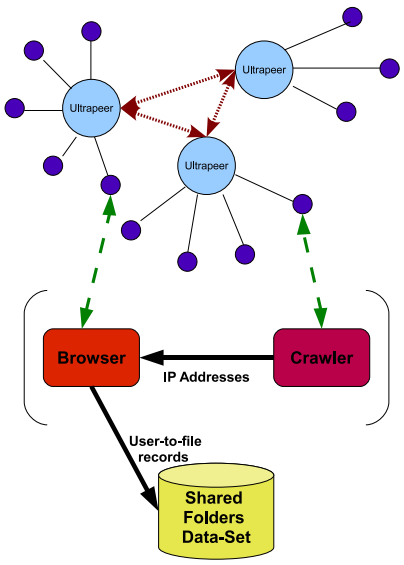

Fig. 1. The Gnutella Crawler-Browser Data Collections System

ties in Section 3. We conclude the paper in Section 4.

2. DATA COLLECTION METHODOLOGY

The Gnutella network is a two-tier topology consisting of a small number of *ultrapeers*, and many *leaves*. Each leaf is connected to one or more ultrapeers. The ultrapeers are connected to each other, and serve as a backbone that facilitates efficient routing of search queries. More details about the Gnutella network can be found in [13, 14, 15, 16].

Collecting shared folder data from Gnutella users is done in two phases. First, one needs to "discover" the current network topology. We call this phase *crawling*. The purpose of the crawling is to generate a list of active Gnutella users. The second phase, called *browsing*, involves querying users for their shared folders data.

Figure 1 depicts the crawling and browsing operations of the system. The crawler treats the network as a graph and performs a breadth-first exploration, where newly discovered nodes are enqueued in a list of un-crawled addresses. 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 [17] 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 network is therefore, never fully covered, and the crawler keeps discovering new IP address. However, when the crawler covers the core of the network, the rate of newly discovered nodes drops. This is an indication that the newly discovered nodes are mostly the ones that have joined the network only after the crawling operation started. A sharp drop in the rate of newly discovered IP addresses, serve as a stopping condition to the crawling operation.

The crawler provide a list of active IP addresses to the browser. The browser sends Gnutella "Query" messages [17] to clients that are currently active. The clients reply with a list of their available shared folder content, which serve as the basis for our p2p based data-set.

2.1. Post-Processing

Post-processing the data involves three main stages. First we sort the data records according to file types. Music related content can be easily identified using the file extension (typically mp3 files). In the Gnutella network, about 70%-80% of the files are music related. The second stage involves geo-identification of the data records. We first generate a list of all the unique IP addresses in the data set (typically over a million). We resolve the geography of IP addresses using the commercial IP2Location[1] database. Each IP address is bounded with its country code, city name, and latitude-longitude values. This geographical information was proved beneficial in community detection [12] and trend prediction [4] tasks.

The IP addresses are a risk to users privacy. The third post-processing stage is thus data-set anonymization. After the geo-identification, there is no need for the actual IP addresses. We thus replace each IP address with a random identifier, which allows for users identification without compromising their privacy.

3. DATA SET STATISTICS

In this section we provide results collected using a 24 hours active crawling of the shared folders of over 1.2 million users on November the 25th 2007. By filtering out musical content (.mp3 files), we identified 531,870 different song files. During the time of the crawl, Gnutella was the most popular file sharing network [18].

Files that had just one digital copy (identified by hash key), were removed from the data-set[2]. Songs were identified using a song id which is the name of the song concatenated by the name of the performing artist. This method account for ambiguities in songs names. Spelling mistakes are handled by grouping together songs with edit-distance smaller than 3 (counting inserts, deletes and substations). This make the method much more resilient to noise since by aggregating the all versions (file hashes) of songs, we increase the content-content links by a quadratic factor. For instance, if song A and song B have 100 common users and A and B have 10 hashes then (assuming uniform dis-

[1] http://www.ip2location.com/

[2] The root cause for this is efficiency since it drastically reduces the size of in-memory content index. However such filtering is also effective in ignoring the single copy files that failed to spread between users.

tribution among file hashes) the strength of each (Ah,Bh) link is just 1.

First we analyze some statistical characterization of the data-set at hand. We then focus on two main aspects that are key in modern search engines and recommender systems: song and user similarity. Song similarity is commonly used in item-based recommendation, and user similarity is used in collaborative filtering for locating like-minded users [19]. In the context of p2p networks, we discuss possible approaches for extracting these similarities while overcoming the inherent "noise" that exists in such networks.

3.1. Shared Songs

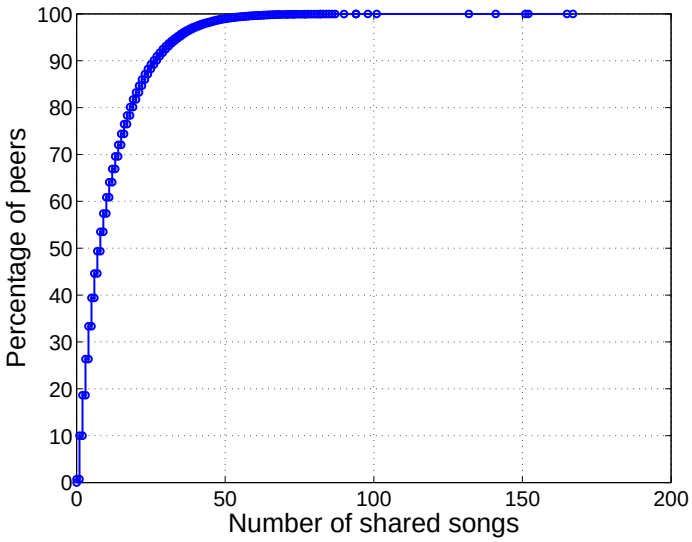

Fig. 2. Songs shared by a sample 100k users, showing cumulative distributions of the number of shared songs

Using a random sample of 100k users, we find the number of different songs each user shares, the maximal overlap (of songs) she has with other users and the percentage of users she has no overlap with. Notice that there are 511k songs in the sampled set, a value which is not much lower than the 530k songs in the original crawl using 1.2 million users. This indicates that most users in the p2p network share similar files. It also suggests that when popular music is considered, exhaustive crawls are not necessary in order to obtain sufficient representative data. However, the "long tail" distribution of the less popular music files, mandates long exhaustive crawls for sufficiently learning the relations of the less popular songs and artists.

Figure 2 depicts the distribution of the number of songs shared by users in our sample. About 85% of the users share less than 20 songs while less than 3% share more than 50 songs. This result matches the observation in [14] regarding "free-riders" in the Gnutella network. Note that all users share less than 200 songs. We attribute this to the finite amount of disk space users are willing to devote for sharing and to the actual amount of different songs that are of interest to a user.

3.2. Metadata

Our data-set include meta-data (ID3 tags) that describe the name of a song, artist, genre and album. This information is often used by search engines when it is matched against relevant query strings. Recommender systems may leverage this meta-data to asses song and user similarity. However, the content in ID3 tags is often missing, misspelled or encoded (e.g., in the genre field). As a result, a recent study [18] showed that only 7%-10% of the queries in the Gnutella network are successful in returning useful content.

Analysis of songs genres reveals that over 35% of the files missing any genre information, while the remaining files have over 3600 different genres. The top genres are Rap (9%) and Rock (8.8%), followed by Pop (4.5%), Country (3.2%), Hip-Hop (2.9%), Blues (2.6%), Soundtrack and Alternative (1.7%), Latin (1.2%) and Metal (1.1%). Similar analysis on artists reveals that 14% of the files are missing an artist tag, while the rest span across over 100,000 different artists. The most common values for "artist" in ID3 tags were *Lil Wayne*, *MC5* and *Jay-Z*. These two fields alone are too coarse and inaccurate to effectively represent songs similarities, or to accurately capture users preferences. We thus focus on methods for extracting users and songs similarities based on p2p information, that will overcome the inherent "noise" and high sparsity of the data-set. Using these techniques it is possible to leverage the advantages of p2p data for improved recommendation systems and search engines.

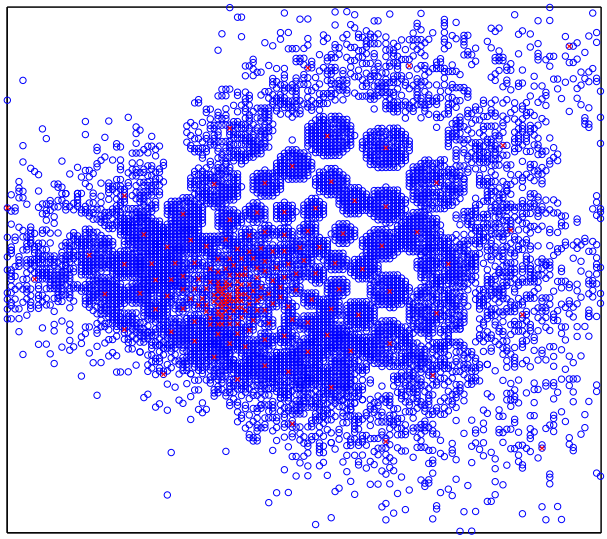

Fig. 3. Embedding and clustering users on a 2-D space (only 200 nearest users to each centroid are shown)

3.3. User Similarity

Estimating user similarity is key for collaborative filtering systems, where the similarity measures are used in order to find like-minded users. We present here two different approaches to achieve the above task in a p2p based data-set. The first method requires associating mp3 files with artists names based on ID3 tags. We then apply *FastMap* embedding [20] followed by k-means [21] clustering. The second methods is based on a technique that was recently suggested by Shavitt *et al.* [12], which does not rely on the existence of any metadata information, but is more computationally intense.

Platt [22] suggested using Fast Sparse Embedding (FSE) to cluster songs in order to extract similarity. FSE is based on projecting a large feature matrix, represented by a graph, into a low dimension space. Similarities are then evaluated by measuring the Euclidian distance of the projected items. We thus used FSE in order to project users into a lower dimension space. First, the "artist" ID3 tag was used in order to construct a user-to-artist graph. We then sampled a subset of 100k users, and pruned the artists vector by taking the top 300 artists for each user.

Figure 3 presents the projected image clustered on a 2-D space. Only the top 200 users nearest to the cluster centroids are shown. While it is possible to identify the distinguished clusters in the periphery, the center of the plot has many overlapping clusters which are quite inseparable. In this figure we used a 2-D projection, which obviously distorts the actual distances. When more dimensions are allowed (in our case $D > 8$), clusters superability improves significantly.

The approach above, has two main limitations. First, it relies on accurate metadata, which is often absent in a p2p data-set. Second, the projection distorts the actual distance between users. As seen in Figure 3, this distortion is most evident when the actual distances are small (in the center), whereas the periphery, that represent much smaller and unique niches, remains less affected. FSE is therefore ineffective in distinguishing between main-stream users.

Overcoming the above limitations is possible by using a technique recently presented by Shavitt *et al.* [12]. The authors suggest using the shared files directly (without additional usage of metadata). Sparsity is handled by using a song similarity graph (presented it in the following section), which enables to calculate the distance between songs (instead of users). Distance between users is estimated using maximal matching, which captures the best overall similarity of shared songs.

Figure 4 compares users similarity measurements based on [12] (x-axis), to users similarity based on artists names in ID3 tags (y-axis). Assuming that two users i and j have two sets of artists A_i and A_j, we define the artist similarity as $(|A_i \cap A_j|) / \min\{|A_i|, |A_j|\}$. Figure 4 shows that there is a high correlation between the techniques, indicating overall correctness. However, for some users, high similarity was observed even when artist similarity is zero, showing once more the down side of using metadata.

Despite these advantages, the technique of Shavitt *et al.* has two main drawbacks. First, it is based on the entire user-to-song graph, a very large graph, which is not easily obtained. Second, the maximal matching algorithm, and the shortest path algorithm (used for songs distances) are computationally intensive.

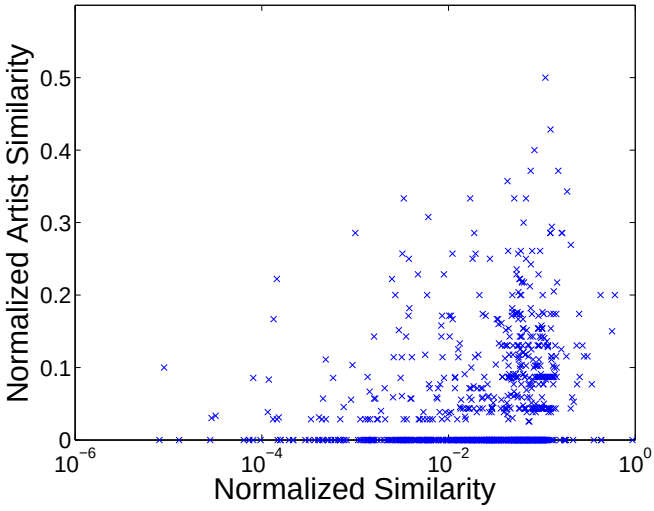

Fig. 4. Comparing user similarity with artist similarity (500 random users are shown for brevity)

3.4. Song Similarity

Songs similarity may be used by item-based recommender systems, where items are recommended based on other related items. In this case, it is not required to find like-minded users, but rather similar items.

A p2p data-set can be modeled as a 2-mode graph that connects users to shared content (songs). This graph is a special case of the standard collaborative filtering matrix in which a link in the graph represents the ranking of an item by a user, whereas in our case, there is no "ranking". The graph can then be collapsed into a 1-mode song similarity graph, where the weight of a link between two songs is the number of users that keep adjacent songs. Additionally, a popularity distribution vector is created, counting the number of times each song appears in the network.

We construct the similarity graph using the complete crawl of 1.2 million users in Gnutella. We prune song links of less than 16 different users, which essentially removes "weak" ties between songs. We then use a second filter that keeps, for each file, only the top 40% links (ordered by descending similarity value) and not less than 10. After these preliminary filters, roughly 20 million undirected links remain.

The degree distribution of the resulting similarity graph is shown in Figure 5. The figure depicts a distinct power-law [23] distribution with a broad set of degrees. The curve observed in the low degrees is attributed to the filtering. This power-law distribution indicates that finding the relationships between many songs in the long tail may require an extensive crawl, while on the other hand, understanding the connectivity of the popular

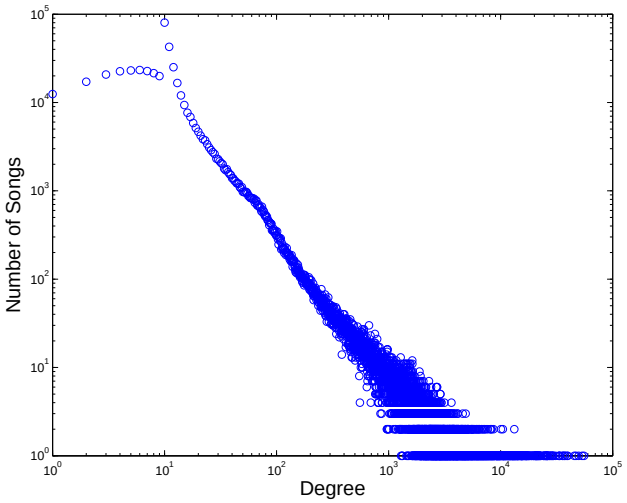

Fig. 5. Degree distribution of the song similarity graph

songs is relatively easy. Recalling that this graph is used for finding distances (or similarity) between songs, it means that popular songs will be better represented in the graph than non-popular songs, hence their distance estimation will be more accurate.

We estimate songs similarity in the same manner as as in [24]. Our approach here is based on song clustering, which is applied directly on the songs similarity graph described above. We ran a modified version of the well known k-means [21] clustering algorithm (using k=100 clusters), which is suited to handle large and sparse graphs. Evaluation of the resulting clusters is performed by finding the dominant genre and artist in each cluster, i.e., the artist and genre that were most frequent (has the highest prevalence). It is expected that the prevalence would be high if the clusters indeed group together similar songs. We found that the median prevalence of the dominant genre is 12%, and over 5% of the clusters, reach over 30% prevalence. This indicates that many of the files in a cluster share common features (recall that there are over 3600 possible genres in our database). Next, we considered the number of additional significant genres in each cluster. A significant genre is defined as the one that has a prevalence of at least half the prevalence of the dominant genre. On average, each cluster contains only 2 significant genres. Recalling that clusters contain thousands of songs, this result demonstrates the correspondence of different songs within the same cluster.

4. SUMMARY

We described a framework for collecting shared folder content from the Gnutella file sharing network. Analysis of the statistical properties of the data-set reveals a power-law structure of the network. However, due to the extreme size of the network and the amount of shared songs, most users have very little overlap with other users. Additionally, the metadata in ID3 tags is partial and often erroneous, making it ineffective in collaborative filtering.

We discuss two efficient techniques for extracting users similarity. The first relies on the existence of metadata, and the second is more computational intense, but does not require any additional information. Finally, we discuss songs similarity based on a song-to-song graph, and use it to cluster similar songs together.

5. REFERENCES

[1] Daniel P. W. Ellis and Brian Whitman, "The quest for ground truth in musical artist similarity," in *Proc. International Symposium on Music Information Retrieval*, 2002.

[2] 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.

[3] Yuval Shavitt and Udi Weinsberg, "Song clustering using peer-to-peer co-occurrences," in *ISM '09: Proceedings of the 2009 11th IEEE International Symposium on Multimedia*.

[4] Noam Koenigstein, Yuval Shavitt, and Tomer Tankel, "Spotting out emerging artists using geo-aware analysis of p2p query strings," in *The 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2008, pp. 937–945, ACM.

[5] Noam Koenigstein and Yuval Shavitt, "Predicting billboard success using data-mining in p2p networks," in *ISM '09: Proceedings of the 2009 11th IEEE International Symposium on Multimedia*.

[6] Noam Koenigstein and Yuval Shavitt, "Song ranking based on piracy in peer-to-peer networks," in *Proc. International Symposium on Music Information Retrieval*, 2009.

[7] Brian McFee and Gert Lanckriet, "Heterogeneous embedding for subjective artist similarity," in *Proc. International Symposium on Music Information Retrieval*, 2009.

[8] Brian Tomasik, Joon Hee Kim, Margaret Ladlow, Malcolm Augat, and Derek Tingle Richard Wicentowski, "Using regression to combine data sources for semantic music discovery," in *International Symposium on Music Information Retrieval*, 2009.

[9] Geraint A. Wiggins, "Semantic gap?? schemantic schmap!! methodological considerations in the scientific study of music," in *ISM '09: Proceedings of the 2009 11th IEEE International Symposium on Multimedia*.

[10] Luke Barrington, Reid Oda, and Gert Lanckriet, "Smarter than genius? human evaluation of music recommender systems," in *International Symposium on Music Information Retrieval*, 2009.

[11] Oscar Celma, *Music Recommendation and Discovery in the Long Tail*, Ph.D. thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2008.

[12] Yuval Shavitt, Ela Weinsberg, and Udi Weinsberg, "Estimating peer similarity using distance of shared files," in *International Workshop on Peer-to-Peer Systems (IPTPS)*, 2010.

[13] Daniel Stutzbach and Reza Rejaie, "Characterizing the two-tier gnutella topology," *SIGMETRICS Performance Evaluation Review*, 2005.

[14] Eytan Adar and Bernardo A. Huberman, "Free riding on gnutella," *First Monday*, vol. 5, 2000.

[15] M. Ripeanu, "Peer-to-peer architecture case study: Gnutella network," in *First International Conference on Peer-to-Peer Computing*, 2001.

[16] Matei Ripeanu, Ian Foster, and Adriana Iamnitchi, "Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design," *IEEE Internet Computing Journal*, vol. 6, 2002.

[17] "The gnutella protocol specification v0.41," 2010, `http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf`.

[18] "Ars technica report on p2p file sharing client market share," 2008, `http://arstechnica.com/old/content/2008/04/study-bittorren-sees-big-growth-limewire-still-1-p2p-app.ars`.

[19] Markus Schedl and Peter Knees, "Context-based Music Similarity Estimation," in *Proceedings of the 3rd International Workshop on Learning the Semantics of Audio Signals (LSAS 2009)*, Graz, Austria, December 2 2009.

[20] Christos Faloutsos and King-Ip Lin, "Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets," in *ACM SIGMOD '95*, 1995.

[21] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. M. Le Cam and J. Neyman, Eds. 1967, vol. 1, pp. 281–297, University of California Press.

[22] John C. Platt, "Fast embedding of sparse music similarity graphs," in *In Advances in Neural Information Processing Systems*, 2004.

[23] Albert-László Barabási and Réka Albert, "Emergence of scaling in random networks," *SCIENCE*, vol. 286, pp. 509 – 512, 1999.

[24] Ela Weinseberg, "Improving searchability in peer-to-peer networks using co-occurrences of shared content," M.S. thesis, Tel-Aviv University, Israel, 2010.