COLLABORATIVE FILTERING BASED ON P2P NETWORKS

Noam Koenigstein[1], Gert Lanckriet[2], Brian McFee[3], and Yuval Shavitt[1]

[1]School of Electrical Engineering, Tel Aviv University
[2]Electrical and Computer Engineering, University of California, San Diego
[3]Computer Science and Engineering, University of California, San Diego

ABSTRACT

Peer-to-Peer (P2P) networks are used by millions of people for sharing music files. As these networks become ever more popular, they also serve as an excellent source for Music Information Retrieval (MIR) tasks. This paper reviews the latest MIR studies based on P2P data-sets, and presents a new file sharing data collection system over the Gnutella. We discuss several advantages of P2P based data-sets over some of the more "traditional" data sources, and evaluate the information quality of our data-set in comparison to other data sources (Last.fm, social tags, biography data, and MFCCs). The evaluation is based on an artists similarity task using Partial Order Embedding (POE). We show that a P2P based Collaborative Filtering data-set performs at least as well as "traditional" data-sets, yet maintains some inherent advantages such as scale, availability and additional information features such as ID3 tags and geographical location.

1. INTRODUCTION

The usage of P2P based information for music information retrieval (MIR) tasks is gaining momentum. The process of collecting Collaborative Filtering (CF) data from P2P networks is typically more complex than from more "traditional" sources such as Last.fm or social networks, but there are several advantages that significantly undermine this small impairment.

Barrington et al. [2] compared different approaches for music recommendation with a user study of 185 subjects. They concluded that approaches based on collaborative filtering which essentially capture the "wisdom of the crowds", outperform content-based approaches so long as the data-set used is sufficiently comprehensive. However, when the data-set is insufficient, or the artists are less popular (those in the long tail), we are compelled to use content-based approaches. The scale of a CF data-set is therefore of great importance. Using a crawler of the Gnutella file-sharing network, we were able to record 281,865,501 user-to-song relations of over 1.3 million users in a single 24 hours crawl. Such scales far exceed the "traditional" CF data-sets such as the well-established Last.fm data-set provided by [6] (17,559,530 records from 359,347 users).

Another advantage of P2P data-sets over traditional data-sets 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 its usage. 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.

Data-sets based on shared folders typically include ID3 tags that reveal information such as the title, artist, album and track number. Although sometimes these records are absent or conflicting, it is often still possible to restore the correct values. In this paper for example, we used majority voting to decide on the correct artist names for different files. P2P data-sets typically include also the IP addresses of the users. The IP address can be used as a unique user identifier for short time spans, but more importantly, it also allows for geographical classification of users. IP-based geographical classification is highly accurate and can reveal not only the user's country and state, but also the user's city and sometimes even smaller areas like the boroughs of New-York City. Such geographical resolution was used by [14] for identifying emerging local musical artists with high potential for a breakthrough.

Despite all their advantages, P2P networks are quite complex, making the collection of a comprehensive data-set 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 content after using it, leaving no trace of its usage.

A different complexity involves the usage of meta-data, which was shown to be particulary useful for finding similarity between performing artists [17]. The content on file sharing networks is mostly ripped by individual users for consumption by other users. User based interactions are a desirable property in IR data-sets, 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 typically

caused by spelling mistakes, missing data, and different variations on the correct values. In the Gnutella network for example, only 7-10% of the queries are successful in returning useful content [30]. A common hash signature can facilitate similar files grouping, nonetheless it does not solve the problem of copies that are not digitally identical. The problem of meta-data ambiguities in P2P data-set is addressed in [16].

2. BACKGROUND

MIR studies based on P2P networks belong to one of two categories:

- **Studies Based on Queries:** Queries in a file-sharing network represent the current tastes and interests of users. A query is issued upon a request by a user searching for a specific file, or content relevant to the search string. Query data-sets are time dependent, and because of dynamic IP assignments, it can be difficult to track a single user over time. Therefore, query-based studies tend to focus on temporal trends such as predicting artists success or artists ranking. Queries are generally ineffective for predicting artist similarity and general recommendation systems because the information gathered per user is limited to a short time period, and thus only a few files per user are usually available.

- **Studies Based on Shared Folders:** The content of a user's shared folder accumulates over time. It can be viewed as an integration of a user's taste over an extended period of time. Data-sets derived from shared folders are therefore the preferred choice for similarity tasks such as recommender systems.

2.1 Previous Query-based Studies

In [14], geographically identified P2P queries were used in order to detect emerging musical talents. The detection algorithm is based on the observation that emerging artists, especially rappers, have a discernible stronghold of fans in their hometown area, where they are able to perform and market their music. In a file-sharing network, this is reflected as a spike in the spatial distribution of queries. The algorithm mimics human scouts by looking for performers which exhibit a sharp increase in popularity within a small geographic region.

The algorithm in [14] is effective for predicting the success of emerging artists, but it cannot be applied on well-established artists. Bhattacharjee et al. [4,5] have used P2P activity to predict an album's life cycle and trends on the Billboard's top 200 albums chart. Both papers used the WinMx file-sharing network. In [4], they showed that P2P sharing activity levels provide leading indicators for the direction of movement of albums on the Billboard charts. In [5], a linear regression model was used to show that sharing activity may be used to predict an album's life cycle. More recently, [17] used the C4.5 [22] and BFTree [8, 26] algorithms on queries collected from the Gnutella network in order to predict a song's top rank on the Billboard singles chart.

A different approach for using P2P queries was taken by [13]. Grace et al. [10] noticed that although music sales are losing their role as means for music dissemination, they are still used by the music industry for ranking artist success, e.g., in the Billboard Magazine chart. They therefore suggested using social networks as an alternative ranking system; a suggestion which is problematic due to the ease of manipulating the list and the difficulty of implementation. Koenigstein et al. [13] used Gnutella queries in order to build an alternative to the Billboard song ranking chart. They compared trends in sales and air-play counts, to piracy popularity trends, and showed that piracy popularity of singles by well-established artists, is highly correlated with the Billboard charts.

2.2 Previous Shared Folders Studies

First attempts to use P2P shared folders for artist similarity were presented in [3, 7]. The centralized and somewhat undersized OpenNap network was used in order to generate a similarity measurement that was based on artists co-occurrences in shared folders. The authors compared the P2P information to other similarity measurements such as social tags in [7], and also Gaussian mixtures over MFCCs and playlists co-occurrences in [3]. The evaluation was done against survey data, and similarities were measured by a pre-determined similarity function.

We took the same approach of evaluating data against a human based survey. The evaluation in this paper is based on the Partial Order Embedding (POE) algorithm of [18], which learns an optimized artist similarity space from labeled (partially ordered) examples. The key difference between the evaluation in [3] and the present work is that we report accuracy achievable by an optimized similarity function, whereas [3] relies on a fixed similarity function. The results in Section 4 emphasize the importance of training the embedding before evaluating with a human based survey. The scale of the data-set used here (13.8 million user-to-song relations after processing), is much higher than in [3,7] (400K user-to-song relations after processing), although our experimental results are restricted to a subset for evaluation purposes.

The first working recommender system based on P2P information was demonstrated in [25]. Shared folders data from the Gnutella network was used in order to generate a user-to-artists matrix. The artists were clustered using k-means algorithm, and recommendations were done from the centroid or from the nearest neighbor.

3. DATA COLLECTION METHODOLOGY

The practice of collecting information from file-sharing networks is relatively common in the field of computer communication. P2P measurement techniques fall into five basic categories:

1. **Passive Monitoring:** Monitoring P2P activity by analyzing data from a gateway router.

2. **Participate:** Developing a client software that can capture and log interesting information [13, 14, 17].

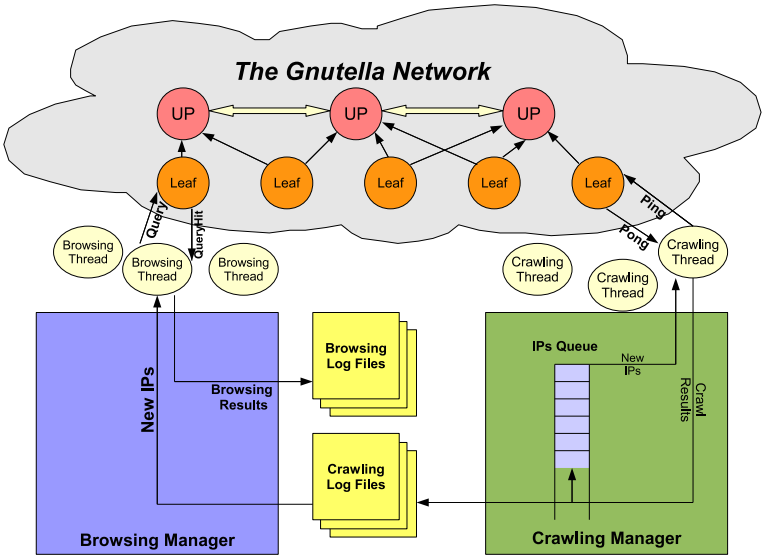

Figure 1. Crawling and Browsing in a Two-Tier Gnutella Segment

3. **Crawl:** Developing a crawler which recursively "walks" the network by asking each peer for a list of its neighbors [15, 16, 25].
4. **Sample:** Sampling a set of peers and gathering static peer properties [4, 5].
5. **Central:** Study information gathered from a central entity in the network [3, 7].

The data collection system described here belongs to the third category. We crawled the Gnutella file-sharing network as described below.

3.1 The Gnutella Network

Gnutella started its operations on March 2000, as the first decentralized file-sharing network. It is arguably the most academically studied file-sharing network [1, 9, 11, 23, 24, 27, 28].In late 2007, it was the most popular file-sharing network on the Internet with an estimated market share of more than 40% [21], serving millions of users.

Modern Gnutella, as well as other popular P2P file-sharing applications, adopted a two-tier topology. In this architecture, a small fraction of nodes, called *ultrapeers*, form an ad-hoc top-level overlay whereas the remaining nodes, called *leaves*, each connect to the overlay through a small number of ultrapeers. Ultrapeers belong to regular users with higher computing and network resources. These nodes route search requests and respond to other users who connected to them. Ultrapeers typically have a high degree (i.e., maintain 30 neighbors) in order to keep a short path lengths between participating peers [28]. We crawl both leaves and ultrapeers in a similar manner.

3.2 Crawling the Network

P2P crawlers operate in a similar way to web crawlers. The crawler treats the network as a graph. The starting points of the crawling operation are taken from an offline initialization list of known hosts. This initialization list must contain some redundancies, because unlike web crawling, the Gnutella nodes might be offline and therefore unresponsive. To maximize the performance of the highly parallelized architecture of the crawler, we used a very large initialization list of 104,767 IP addresses. This allows us to make use of all the crawling clients right at the beginning of the crawling operation [1] .

Figure 1 depicts the crawling and browsing operations in a two-tier Gnutella segment. The crawling process is a breadth-first exploration, where newly discovered leaves and ultrapeers are enqueued in a list of un-crawled addresses (The *IPs Queue*). The parallel crawling threads constantly ask the *Crawling Manager* for new IP addresses from the queue, and send back newly received results. The results are stored in text log files, and new IPs are enqueued in the IPs Queue.

Gnutella's "Ping-Pong" protocol is used by the crawling threads to discover new Gnutella nodes in the network. A node receiving a "Ping" message is expected to respond with one or more "Pong" messages. A "Pong" message includes the address of a connected Gnutella node and information regarding the amount of data it is making available to the network. An incoming Ping message with TTL = 2 and Hops = 0 is a "Crawler Ping" used to scan the network. It should be replied to with Pongs containing information about the node receiving the Ping and all other nodes it is connected to. More details about the the Gnutella protocol can be found in [29].

The crawling of large scale *dynamic* networks, such as file-sharing networks never reaches a full stop. As clients constantly connect and disconnect from the network, the crawler will always discover new IP addresses. We thus use two stopping conditions: A time constraint (typically 1 hour), or reaching a low rate of newly discovered nodes, which indicates the completion of a crawl. In the beginning of a crawl, the rate of newly discovered nodes increases dramatically and typically reaches over 300,000 new clients per minute. As the crawling process proceeds,

[1] Such a large list of IP addresses can be easily generated from the results of a previous crawling operation.

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.

3.3 Browsing Shared Folders

The browsing operation begins shortly after the crawling operation started. Once the first crawling log file is created, the *Browsing Manager* can start assigning IP addresses (taken from the crawling logs) to the browsing threads. The browsing threads send "Query" messages to the Gnutella nodes, and wait for a "QueryHit" message in return. Query messages with TTL=1, hops=0 and Search Criteria="␣␣␣␣" (four spaces) are used to index all files a node is sharing. A node should reply to such queries with all of its shared files. The sharing information is stored by the *Browsing Manager* in the browsing logs. These files are used to generate the CF data.

4. EVALUATION

To evaluate the information content of the present P2P data, we test its performance on an artist similarity prediction task. [18,19] developed the Partial Order Embedding (POE) algorithm for integrating multiple data sources to form an optimized artist similarity space, and applied it to acoustic models (Gaussian mixtures over MFCCs and chroma vectors), semantic models (semantic multinomial auto-tags, and social tags from Last.fm [2]), and text models (biography data) [3]. By applying the same algorithm to collaborative filtering data, we can evaluate the amount of high-level artist similarity information captured by P2P collaborative filtering data, and quantitatively compare it to alternative data sources.

In general, collaborative filtering has been repeatedly demonstrated to be an effective source of information for recommendation tasks (see, e.g., [2, 12]). One may then wonder how one source of collaborative filtering data compares to another. Because [18] did not include collaborative filtering in their experiments, there is no existing baseline to compare against for the artist similarity task. We therefore repeat the experiment with the Last.fm collaborative filtering data provided by [6], allowing us to quantitatively compare P2P data to a more conventional source of collaborative filtering information.

We sampled 100K (7.69%) users out of over 1.3 million Gnutella users recorded on a single 24 hours crawl. We filtered the files that correspond to musical files according to file suffix (.mp3 files). In the entire (1.3 million users) data set, we identified 531,870 different songs. In our 100K users sample, we identified 511K songs, a value that is not much lower than the total number. Ambiguities in artist names due to typos and misspellings were corrected by majority voting. After this step, we had 13,839,361 user-to-song relations, which was the base for a collaborative matrix (ARTISTSxUSERS). The artist names are the same as in the aset400 data set of Ellis and Whitman [7]. These

[2] http://last.fm/

[3] The data from [18] can be found at http://mkl.ucsd.edu/.

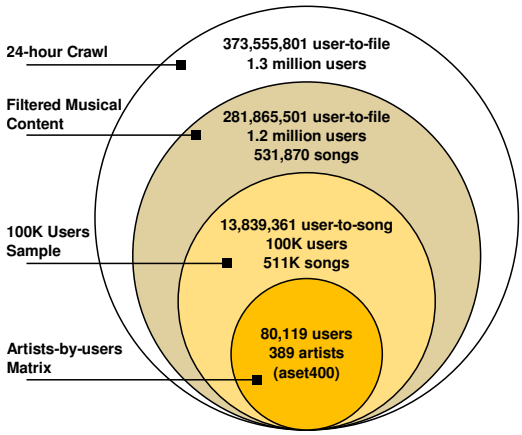

Figure 2. A quantitative summary of the data-set scale after each processing stage

artists were found in the shared folders of 80,119 users. The above numbers are summarized in Figure 2. Our similarity matrix will be available on the authors website by publication time.

4.1 The embedding problem

Formally, the goal in this experiment is to learn an embedding function $g : \mathcal{X} \rightarrow \mathbb{R}^n$, which maps a set of artists $\mathcal{X}$ into Euclidean space. The embedding is trained to reproduce relative comparison measurements (i, j, k), where (i, j) are more similar to each-other (i.e., closer) than (i, k).

Each artist is represented as a vector in some feature space, and the embedding function is parameterized as a linear projection from that feature space to the embedding space. This can be expressed in terms of inner products:

$$g(i) = NK_i,$$

where N is a linear projection matrix to be learned, and K_i is a vector containing the inner product of i's feature vector with each other point in the training set. As described in [18], this readily generalizes to non-linear kernel functions and heterogeneous data sources, but we do not make use of these extensions in the present experiment.

To summarize, given a set of training artists, relative similarity measurements between the artists, and a feature representation of each artist (equivalently, a kernel matrix over the training artists), the algorithm finds a linear projection matrix N which attempts to satisfy the similarity measurements under Euclidean distance calculations:

$$(i, j, k) \Leftrightarrow \|N(K_i - K_j)\| < \|N(K_i - K_k)\|.$$

The matrix N is found by solving a convex optimization problem, which involves three competing terms:

$$\max_W \sum_{i,j} \|K_i - K_j\|_W^2 - \beta \cdot \sum_{ijk} \xi_{ijk} - \gamma \cdot \mathrm{tr}(WK)$$
$$\|K_i - K_j\|_W^2 \doteq (K_i - K_j)^\mathsf{T} W (K_i - K_j),$$

where W is a positive semi-definite matrix which can be factored to recover the projection matrix: $W = N^\mathsf{T} N$.

The first term maximizes the variance of the data in the embedding space, which prevents points from being collapsed onto each-other.

The second term tries to minimize the number of ordering mistakes made by the embedding function. This is accomplished by using a slack variable $\xi_{ijk} \geq 0$ for each triplet constraint (as in support vector machines), allowing for margin violations:

$$\|K_i - K_j\|_W^2 \leq \|K_i - K_k\|_W^2 + 1 - \xi_{ijk}.$$

Finally, the third term limits the complexity of the learned space by penalizing a convex approximation to the rank of the embedding space. For more details about the optimization procedure, see [18].

At test time, similarity queries are presented in a similar form: (q, i, j), where q is previously unseen, and i and j come from the training set. The query artist is mapped into the embedding space by first computing inner products to the training set, resulting in a vector K_q, and then projecting by N: $g(q) = NK_q$. Once in the embedding space, distances are calculated to i and j, and the similarity prediction is counted as correct if the distance to i is smaller than the distance to j.

4.2 From P2P to artist similarity

In order to apply the POE algorithm to collaborative filtering data, we need to define a kernel function between artists in terms of the collaborative filtering matrix. One straightforward choice of kernel function is to simply count the number of users shared between two artists i and j. However, this may suffer from popularity bias if i has many users and j has relatively few. To counteract this, we normalize each artist by the number of users to which it is matched. This gives rise to the kernel function:

$$k(i,j) = \frac{\#\text{users for } i \text{ and } j}{(\#\text{users for } i) \cdot (\#\text{users for } j)}.$$

Equivalently, we can interpret this kernel function as the cosine-similarity between *bag-of-users* representations of artists i and j, i.e., an artist is represented by a binary vector where coordinate z is 1 if user z is present and 0 otherwise. This is similar to the bag-of-words representation commonly used in text applications, and like in text, the dimensionality of the feature representation is much larger than the number of data points (i.e., there are many more users than artists). Consequently, it is more economical to use the kernel matrix representation than to work directly on the feature vectors.

4.3 Results

We reproduced the main experiment of [18], using P2P collaborative filtering data, as well as listener data from Last.fm [6]. We first pruned both data sets down to the 412 artists of aset400 [7]. Of these artists, 23 were missing from P2P, and 5 were missing from Last.fm. Nonetheless, we retain similarity measurements for these artists to maintain comparability with the previously published results.

As in [18], the artists (and corresponding similarity measurements) are split by 10-fold cross-validation, and the training and test procedure is repeated for each fold. We then calculate the accuracy of the learned embeddings, averaged across all folds. Results are presented in Table 1.

Data source	Native	Learned	Restricted
P2P	0.561	0.728	0.741
Last.fm	0.570	0.760	0.763
MFCC	0.535	0.620	
Biography	0.514	0.705	
Tags	0.705	0.776	

Table 1. Test accuracy for artist similarity. *Native* corresponds to similarity measurements taken from the raw kernel matrix, and *learned* corresponds to similarities learned by POE. The *restricted* column reports accuracy achieved by testing only on artists observed in the data (389 artists for P2P and 407 for Last.fm). See Section 4.3 for details.

The accuracy of similarity predictions may be skewed due to testing on artists for which the data source may have no information (i.e., no users shared songs by that artist). To quantify this effect, we also computed accuracy on similarity measurements restricted to include only those artists observed in collaborative filtering data. These results are given in the *restricted* column of Table 1.

Overall, Table 1 indicates that both P2P and Last.fm collaborative filtering data captures a great deal of high-level artist similarity information. Both sources perform comparably to highly detailed social tags (Tags), and both outperform similarity models derived from artist biographies (Biography) or acoustic content (MFCCs) as reported in [18].

In this experiment, the Last.fm data achieves slightly higher accuracy than the P2P data. However the difference is quite small, and might be eliminated by using a larger sample of P2P users (we only used 7.69%). Also note that the results dramatically improve once the embedding is trained. This emphasizes the importance of learning an optimal similarity space, rather than using a pre-determined similarity function as in [3].

5. SUMMARY

We reviewed the latest P2P based MIR studies, and presented a new Gnutella-based data collection system. We evaluated the information content of our P2P data-set on an artist similarity prediction task based on the Partial Order Embedding (POE) presented in [18], and compared it to the "traditional" data sources, such as Last.fm collaborative filtering, tags, and acoustic models. We showed that a P2P based Collaborative Filtering data-set performs comparably to "traditional" data-sets, yet maintains some inherent advantages such as scale, availability and additional information features such as ID3 tags and geographical location.

According to the International Federation of the Phonographic Industry (IFPI) 95% of all music is downloaded in file sharing networks [20]. We expect that as the practice of file-sharing becomes even more widespread, the usage of P2P based data-sets will become increasingly relevant.

6. REFERENCES

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

[2] Luke Barrington, Reid Oda, and Gert Lanckriet. Smarter than genius? human evaluation of music recommender systems. In *International Symposium on Music Information Retrieval*, 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] Sudip Bhattacharjee, Ram Gopal, Kaveepan Lertwachara, and James R. Marsden. Whatever happened to payola? an empirical analysis of online music sharing. *Decis. Support Syst.*, 42(1):104–120, 2006.

[5] Sudip Bhattacharjee, Ram D. Gopal, Kaveepan Lertwachara, and James R. Marsden. Using P2P sharing activity to improve business decision making: proof of concept for estimating product life-cycle. *Elec. Commerce Research and Applications*, 4(1):14–20, 2005.

[6] O. Celma. *Music Recommendation and Discovery in the Long Tail*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2008.

[7] Daniel P. W. Ellis and Brian Whitman. The quest for ground truth in musical artist similarity. In *International Symposium on Music Information Retrieval*, pages 170–177, 2002.

[8] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression : A statistical view of boosting. *Annals of statistics*, 28(2):337–407, 2000.

[9] Adam Shaked Gish, Yuval Shavitt, and Tomer Tankel. Geographical statistics and characteristics of p2p query strings. In *International Workshop on Peer-to-Peer Systems*, February 2007.

[10] J. Grace, D. Gruhl, K. Haas, M. Nagarajan, C. Robson, and N. Sahoo. Artist ranking through analysis of online community comments. *International World Wide Web Conference*, 2008.

[11] Mihajlo A. Jovanovic. Modeling large-scale peer-to-peer networks and a case study of gnutella. Master's thesis, University of Cincinatti, Cincinatti, OH, USA, 2001.

[12] J. Kim, B. Tomasik, and D. Turnbull. Using artist similarity to propagate semantic information. In *Proc. International Symposium on Music Information Retrieval*, 2009.

[13] Noam Koenigstein and Yuval Shavitt. Song ranking based on piracy in peer-to-peer networks. In *International Symposium on Music Information Retrieval*, Kobe, Japan, October 2009.

[14] 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*, pages 937–945, Las Vegas, NV, USA, 2008.

[15] Noam Koenigstein, Yuval Shavitt, Tomer Tankel, Ela Weinsberg, and Udi Weinsberg. A framework for extracting musical similarities from peer-to-peer networks. In *IEEE International Conference on Multimedia and Expo (ICME 2010)*, Singapore, July 2010.

[16] Noam Koenigstein, Yuval Shavitt, Ela Weinsberg, and Udi Weinsberg. On the applicability of peer-to-peer data in music information retrieval research. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, Utrecht, the Netherlands, August 2010.

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

[18] Brian McFee and Gert Lanckriet. Heterogeneous embedding for subjective artist similarity. In *International Symposium on Music Information Retrieval*, Kobe, Japan, October 2009.

[19] Brian McFee and Gert R.G. Lanckriet. Partial order embedding with multple kernels. In *Proceedings of the 26th annual International Conference on Machine Learning (ICML)*, pages 721–728, 2009.

[20] IFPI: International Federation of the Phonographic Industry. Digital music report 2009.

[21] Ars Technica Report on P2P File Sharing Client Market Share. `http://arstechnica.com/old/content/2008/04/study-bittorren-sees-big-growth-limewire\-still-1-p2p-app.ars`, 2008.

[22] J. R. Quinlan. Learning with continuous classes. pages 343–348, 1992.

[23] Amir H. Rasti, Daniel Stutzbach, and Reza Rejaie. On the long-term evolution of the two-tier gnutella overlay. In *IEEE Global Internet Symposium*, Barcelona, Spain, April 2006.

[24] 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*, 6, 2002.

[25] 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*, December 2009.

[26] Haijian Shi. Best-first decision tree learning. Master's thesis, University of Waikato, Hamilton, NZ, 2007. COMP594.

[27] K. Sripanidkulchai. The popularity of gnutella queries and its implications on scalability, February 2001. Featured on O'Reilly's www.openp2p.com website.

[28] Daniel Stutzbach and Reza Rejaie. Characterizing the two-tier gnutella topology. *SIGMETRICS Perform. Eval. Rev.*, 33(1):402–403, 2005.

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

[30] Matei A. Zaharia, Amit Chandel, Stefan Saroiu, and Srinivasan Keshav. Finding content in file-sharing networks when you can't even spell. In *IPTPS*, 2007.