The LFM-1b Dataset for Music Retrieval and Recommendation

Markus Schedl
Department of Computational Perception
Johannes Kepler University
Linz, Austria
markus.schedl@jku.at

ABSTRACT

We present the LFM-1b dataset of more than one billion music listening events created by more than 120,000 users of Last.fm. Each listening event is characterized by artist, album, and track name, and further includes a timestamp. On the (anonymous) user level, basic demographics and a selection of more elaborate user descriptors are included.

The dataset is foremost intended for benchmarking in music information retrieval and recommendation. To facilitate experimentation in a straightforward manner, it also includes a precomputed user-item-playcount matrix. In addition, sample Python scripts showing how to load the data and perform efficient computations are provided. An implementation of a simple collaborative filtering recommender rounds off the code package.

We discuss in detail the LFM-1b dataset's acquisition, availability, statistics, and content, and place it in the context of existing datasets. We also showcase its usage in a simple artist recommendation task, whose results are intended to serve as baseline against which more elaborate techniques can be assessed. The two unique features of the dataset in comparison to existing ones are (i) its substantial size and (ii) a wide range of additional user descriptors that reflect their music taste and consumption behavior.

Keywords

Dataset, Analysis, Music Information Retrieval, Music Recommendation, Collaborative Filtering, Experimentation

1. MOTIVATION

Research and development in music information retrieval (MIR) and music recommender systems has seen a sharp increase during the past few years, not least due to the proliferation of music streaming services [8, 5]. Having tens of millions of music pieces available at the listeners' fingertips requires novel retrieval, recommendation, and interaction techniques for music.

ICMR'16, June 06 - 09, 2016, New York, NY, USA

ISBN 978-1-4503-4359-6/16/06…$15.00

DOI: http://dx.doi.org/10.1145/2911996.2912004

However, researchers interested in conducting experiments in music retrieval and recommendation on a large scale — in particular those working in academia — are facing the challenge that they frequently have to acquire the data for their experiments themselves, which results in non-standardized collections, hindering reproducibility . While online music platforms such as Spotify,[1] Last.fm,[2] or Soundcloud[3] offer convenient API endpoints that provide access to their databases, it takes a considerable amount of time to build collections of substantial size necessary for large-scale evaluation. The few publicly available datasets for these purposes, the most well-known of which is probably the *Million Song Dataset* (MSD) [2], might represent an alternative, but come with certain restrictions. For instance, while the MSD offers a great variety of pieces of information (among others, genre labels, tags, term weights of lyrics, song similarity information, and aggregated playcount data), user- and listening-specific information is provided rather scarcely, on a high level, or in a summarized form only.

Since we believe that listener-specific information is key to build personalized music retrieval systems, the focus and unique feature of the LFM-1b dataset presented here is detailed information on the level of listeners and of listening events. For instance, the dataset provides user-specific scores about their music taste (among others, measures of mainstreaminess and inclination to listen to novel music). Next to this, the size of the dataset, which includes more than a billion listening events of approximately 120,000 users should be sufficient to perform experimentation on a large scale on real-world data.

The paper is organized as follows. We first review related datasets for music retrieval and recommendation (Section 2). Subsequently, we outline the data acquisition procedure, present the dataset's structure and content, provide basic statistics and analyze them, and point to sample Python scripts that show how to access the components of the dataset (Section 3). We further illustrate how to exploit the dataset for the use case of building a music recommender system that implements various recommendation algorithms (Section 4). We eventually round off the paper with a summary and discussion of possible extensions (Section 5).

2. RELATED WORK

The need for user-aware and multimodal approaches to music retrieval and recommendation has been acknowledged

[1] https://developer.spotify.com/web-api
[2] http://www.last.fm/api
[3] https://developers.soundcloud.com/docs/api/guide

many times and is meanwhile widely accepted [9, 11, 14, 15]. However, respective scientific work is still in its fledgling stage. One of the reasons for this is that involving users, which is an obvious necessity to build user-aware approaches, is time-consuming and hardly feasible on a large scale — at least not in academia. As a consequence, datasets offering user-specific information are scarce.

On the other hand, thanks to evaluation campaigns in the fields of music information retrieval and music recommendation, including the *Music Information Retrieval Evaluation eXchange*[4] (MIREX) and the *KDD Cup 2011*[5] [4], the research community has been given several datasets that can be used for a wide range of MIR tasks, from tempo estimation to melody extraction to emotion classification. Most of these datasets, however, are specific to a particular task, e.g., onset detection or genre classification. What is more, for content-based or audio-based approaches, the actual audio can typically not be shared, because of restrictions imposed by intellectual property rights.

Datasets that can be used to some extent for evaluating personalized approaches to music retrieval and recommendation include the *Yahoo! Music* dataset [4], which currently represents the largest available music recommendation dataset, including more than 262 million ratings of more than 620 thousand music items created by more than one million users. The ratings cover a time range from 1999 to 2010. However, the dataset is completely anonymized, i.e., not only users, but also items are unknown. The absence of any descriptive metadata and ignorance of music domain knowledge therefore restricts the usage of the dataset to rating prediction and collaborative filtering [13].

The *Million Song Dataset*[6] (MSD) [2] is perhaps one of the most widely used datasets in MIR research. It offers a wealth of information, among others, audio content descriptors such as tempo, key, or loudness estimates, editorial item metadata, user-generated tags, term vector representations of lyrics, and playcount information. While the MSD provides a great amount of information about one million songs, it has also been criticized, foremost for its lack of audio material, the obscurity of the approaches used to extract content descriptors, and the improvable integration of the different parts of the dataset. The *MSD Challenge*[7] [10] further increased the popularity of the dataset. Organized in 2012, the goal was to predict parts of a user's listening history, given another part.

Providing more than one million temporally and spatially annotated listening events that have been extracted from microblogs, the *Million Musical Tweets Dataset*[8] (MMTD) [6] particularly supports context-aware recommendation [1]. Each listening event is accompanied by longitude and latitude values, as well as month and weekday. A major shortcoming of this dataset is its uneven geographical distribution of listening events, which is caused by the likewise skewed distribution of microblogging activity around the world.

Another related dataset is constituted of Last.fm data provided by Celma [3]. The dataset comprises two subsets, one containing listening information for about 360 thousand users, only including artists they most frequently listened to. The other subset offers full listening data of nearly a thousand users, where each listening event is annotated with a timestamp, artist, and track name. Both subsets include gender, age, country, and date of registering at Last.fm, as provided by their API.

Other datasets that are related to LFM-1b to a smaller extent include the *AotM-2011* dataset of playlists extracted from Art of the Mix[9] and the *MagnaTagATune*[10] dataset [7] of user-generated tags and relative similarity judgments between triples of tracks.

In comparison to the datasets most similar to the one proposed here — the MSD and Celma's [3] — the LFM-1b dataset offers the following unique features: (i) substantially more listening events, i.e., over one billion, in comparison to roughly 48 and 19 million, respectively, for MSD and Celma's [3]; (ii) exact timestamps of each listening event, unlike MSD; (iii) demographic information about listeners in an anonymous way, unlike MSD; and (iv) additional information describing the listeners' music preferences and consumption behavior, unlike both MSD and Celma's [3]. These additional descriptors include temporal aspects of listening behavior as well as novelty and mainstreaminess scores as proposed in [12], among others.

3. THE LFM-1B DATASET

In the following, we outline the data acquisition procedure from Last.fm, describe in detail the dataset's components, analyze basic statistical properties of the dataset, provide download links, and refer to some sample code in Python, which is also available for download. Please note that the LFM-1b dataset is considered derivative work according to paragraph 4.1 of Last.fm's API Terms of Service.[11]

3.1 Data Acquisition

We first use the overall 250 top tags[12] to gather their top artists[13] using the Last.fm API. For these artists, we fetch the top fans, which results in 465,000 active users. For a randomly chosen subset of 120,322 users, we then obtain their listening histories.[14] For approximately 5,000 users, we cap the fetched listening histories at 20,000 listening events in order to avoid ending up with an extraordinarily uneven user distribution (cf. Section 3.3), in which a few users have an enormous amount of listening events. We define a listening event as a quintuple specified by user, artist, album, track, and timestamp. The period during which we fetched the data ranges from January 2013 to August 2014.

3.2 Dataset Availability and Content

The whole LFM-1b dataset of approximately 8 GB can be downloaded from www.cp.jku.at/datasets/LFM-1b. For ease of access and compatibility, the metadata on artists, albums, tracks, users, and listening events is stored in simple text files, encoded in UTF-8, while the user-artist-playcount matrix is provided as sparse matrix in a Matlab file, which

[4]http://www.music-ir.org/mirex/wiki
[5]http://www.sigkdd.org/kdd2011/kddcup.shtml
[6]http://labrosa.ee.columbia.edu/millionsong
[7]http://www.kaggle.com/c/msdchallenge
[8]http://www.cp.jku.at/datasets/MMTD
[9]http://www.artofthemix.org
[10]http://mi.soi.city.ac.uk/blog/codeapps/the-magnatagatune-dataset
[11]http://www.last.fm/api/tos
[12]http://www.last.fm/api/show/tag.getTopTags
[13]http://www.last.fm/api/show/tag.getTopArtists
[14]http://www.last.fm/api/show/user.getRecentTracks

Table 1: Description of the files constituting the LFM-1b dataset. Attributes of same color are connected to each other.

File	Content
LFM-1b_users.txt	user-id, country, age, gender, playcount, registered_timestamp
LFM-1b_users_additional.txt	user-id, novelty_artist_avg_month, novelty_artist_avg_6months, novelty_artist_avg_year, mainstreaminess_avg_month, mainstreaminess_avg_6months, mainstreaminess_avg_year, mainstreaminess_global, cnt_listeningevents, cnt_distinct_tracks, cnt_distinct_artists, cnt_listeningevents_per_week, relative_le_per_weekday1, ... relative_le_per_weekday7, relative_le_per_hour0, ... relative_le_per_hour23
LFM-1b_artists.txt	artist-id, artist-name
LFM-1b_albums.txt	album-id, album-name, artist-id
LFM-1b_tracks.txt	track-id, track-name, artist-id
LFM-1b_LEs.txt	user-id, artist-id, album-id, track-id, timestamp
LFM-1b_LEs.mat	idx_users (vector), idx_artists (vector), LEs (sparse matrix)

Table 2: Description of the additional user features on preference and consumption behavior.

Attribute	Description
user-id	user identifier
novelty_artist_avg_month	novelty score according to [12], i.e., percentage of new artists listened to, averaged over time windows of 1 month
novelty_artist_avg_6months	novelty score, averaged over time windows of 6 months
novelty_artist_avg_year	novelty score, averaged over time windows of 12 months
mainstreaminess_avg_month	mainstreaminess score according to [12], i.e., overlap between the user's listening history and an aggregate listening history of all users, averaged over time windows of 1 month
mainstreaminess_avg_6months	mainstreaminess score, averaged over time windows of 6 months
mainstreaminess_avg_year	mainstreaminess score, averaged over time windows of 12 months
mainstreaminess_global	mainstreaminess score, computed for the entire period of the user's activity on Last.fm
cnt_listeningevents	total number of the user's listening events (playcounts) included in the dataset
cnt_distinct_tracks	number of unique tracks listened to by the user
cnt_distinct_artists	number of unique artists listened to by the user
cnt_listeningevents_per_week	average number of listening events per week
relative_le_per_weekday[1–7]	fraction of listening events for each weekday (starting on Monday) among all weekly plays, averaged over the user's entire listening history
relative_le_per_hour[0–24]	fraction of listening events for each hour of the day (starting with the time span 0:00-0:59) among all 24 hours, averaged over the user's entire listening history

complies to HDF5 format. This makes the matrix also accessible from a wide range of programming languages. For instance, Python code for data import is provided along with the dataset, cf. Section 3.4.

Table 1 gives an overview of the dataset's content, in particular the included files and respective pieces of information. Keys that are linked to each other are depicted in the same color. Files `LFM-1b_artists.txt`, `LFM-1b_albums.txt`, and `LFM-1b_tracks.txt` contain the metadata for artists, albums, and tracks, respectively. File `LFM-1b_LEs.txt` contains all listening events, described by user, artist, album, and track identifiers. Each event is further attached a timestamp, which is encoded in Unix time, i.e., seconds since January 1, 1970 (UTC). File `LFM-1b_LEs.mat` contains the user-artist-playcount matrix (UAM) as Matlab file in HDF5 format. It comprises 3 items: (i) a 120,175-dimensional vector (idx_users), each element of which links to the user-ids in files `LFM-1b_users.txt`, `LFM-1b_users_additional.txt`, and `LFM-1b_LEs.txt`, (ii) a 585,095-dimensional vector (idx_artists), whose elements link to the artist-ids in `LFM-1b_LEs.txt` and the metadata files, and (iii) a 120,175 × 585,095 sparse matrix (LEs), whose rows correspond to users and columns to artists. User-specific information is given in `LFM-1b_users.txt` and `LFM-1b_users_additional.txt`. While the former contains basic demographic information as well as overall playcount and date of registration with Last.fm, the latter provides 43 additional user descriptors that represent a unique feature of LFM-1b. Table 2 describes these user features, which are valuable in particular when creating user-aware music recommender systems.

3.3 Dataset Statistics

Table 3 shows basic quantitative statistics of the dataset's composition. The number of unique <user, artist> pairs corresponds to the number of entries in the UAM, which is a 120,175 × 585,095 sparse matrix. Note that these numbers are smaller than the total numbers of unique users and artists reported in Table 3 since we discarded users who listened to less than 10 unique artists and artists listened to by less than 10 users when creating the UAM. We assume that data about these artists and users is too sparse to be informative, or just noise. In particular, this approach efficiently filters artists that are misspelled, which is evidenced by the substantial reduction of their number by 81.66% (from 3,190,371 to 585,095). The reduction in terms of users is much smaller (by 0.21%, from 120,322 to 120,175), because users with such a narrow music artist taste are almost non-existent on Last.fm. This filtering step yields a UAM that is very well manageable with today's computers (approximately 200 MB).

Table 3: Statistics of items in the dataset.

Item	Number
Users	120,322
Artists	3,190,371
Albums	15,991,038
Tracks	32,291,134
Listening events	1,088,161,692
Unique <user, artist> pairs	61,534,450

In the following, we present a more detailed analysis of the demographic coverage, distribution of listening events, and features related to music preference and consumption behavior.

Demographics

We compute and illustrate the distribution of users among country, age, and gender. Table 4 shows the countries where most users in the dataset originate from. We include all countries with more than 1,000 users. As can be seen, a majority of users do not provide country information (54.13%). The country-specific percentages in the last column of the table are computed only among those users who provide their country. The distribution of users in the dataset reflects that of Last.fm users in general.

A histogram illustrating the age distribution is shown in Figure 1. Among all users, only 38.31% provide this piece of information. It can be seen that the age distribution is quite uneven and skewed towards the right (higher ages), but reflects the composition of Last.fm users. In addition to this, we can spot some seemingly erroneous information provided by some users, i.e., 165 of them indicated an age smaller or equal to 6 years, 149 indicated an age of at least 100 years. However, the share of these users only represents 0.26% of all users in the dataset. The age distribution has its arithmetic mean at 25.4 years, standard deviation of 9.7, a median of 23, and 25- and 75-percentile, respectively, at 20 and 28 years.

Table 5 depicts the gender distribution of users in the dataset. Among those who provide this information, more than two thirds are male, less than one third female. The larger share of male users on Last.fm is a known fact. The number of users who provide information on their gender (64,551 or 53.6%) is very close to the number of users who provide country information (65,132 or 54.1%), and considerably higher than the amount of users who indicate their age (46,095 or 38.3%). Therefore, users seem to be highly reluctant to reveal their age.

Listening events

To gain an understanding of the distribution of listening events in the dataset, Figures 2 and 3 illustrate the sorted amount of listening events for all artists and for all users, respectively, plotted as red lines. The blue plots indicate the number of listeners each artist has (Figure 2) and the number of artists each user listens to (Figure 3). The axes in both figures are logarithmically scaled.

From Figure 2, we observe that especially in the range of artists with extraordinarily high playcounts (left side of the figure), the number of playcounts decreases considerably faster than the number of listeners. For instance, the top-played artist is on average listened to 78.92 times per user,

Table 4: Statistics on country distribution of users. All countries with more than 1,000 users are shown.

Country	No. of users	Pct. in dataset
US	10255	18.581 %
RU	5024	9.103 %
DE	4578	8.295 %
UK	4534	8.215 %
PL	4408	7.987 %
BR	3886	7.041 %
FI	1409	2.553 %
NL	1375	2.491 %
ES	1243	2.252 %
SE	1231	2.230 %
UA	1143	2.071 %
CA	1077	1.951 %
FR	1055	1.912 %
N/A	65132	54.131 %

Table 5: Statistics on gender distribution of users.

Gender	No. of users	Pct. in dataset
Male	39969	71.666 %
Female	15802	28.334 %
N/A	64551	53.649 %

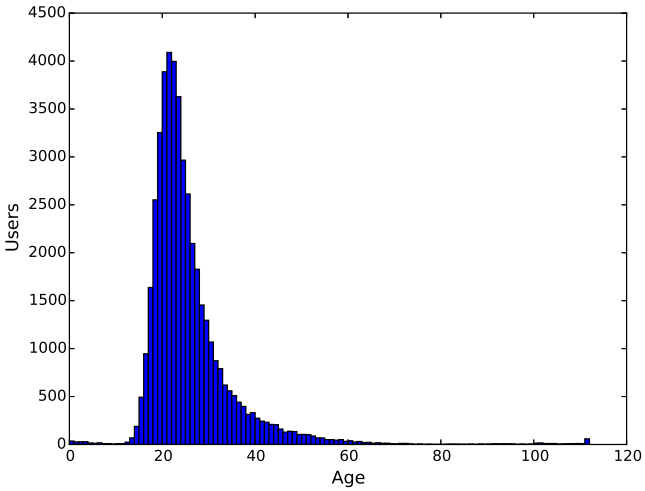

Figure 1: Histogram of age distribution.

while the 1,000$^{\text{th}}$ most popular artist is listened to only 22.66 times per user, on average. On the other side, the 100,000 least popular artists are played only 1.99 times on average. This provides strong evidence of the "long tail" of artists [3].

From Figure 3, we see that highly active listeners (in the left half of the figure) tend to have a rather stable relationship between total playcounts and number of artists listened to, whereas the average number of playcounts per artist strongly decreases for less active listeners. Indeed, the 1,000 most active listeners aggregate on average 29.73 listening events per artist, while for the 1,000 least active listeners, this number is only 3.04. Therefore, highly active users tend to listen to tracks by the same artists over and over again, while occasional and seldom listeners tend to play only a few tracks by their preferred artists. Furthermore, we can ob-

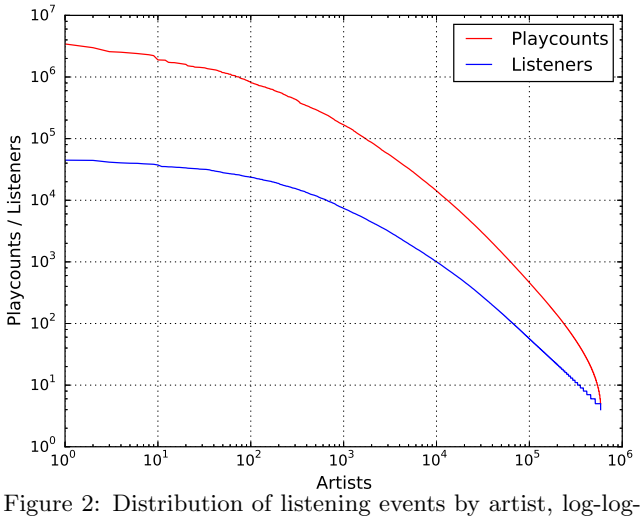

Figure 2: Distribution of listening events by artist, log-log-scaled.

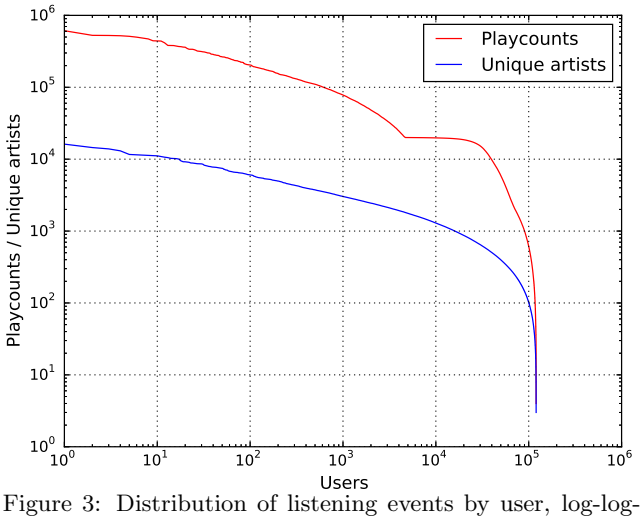

Figure 3: Distribution of listening events by user, log-log-scaled.

serve in Figure 3 the considerable number of users for which we recorded approximately 20,000 listening events, for the reasons given in Section 3.1.

Table 6 shows additional statistics of the listening event distribution, both from a user and an artist perspective (second and third column, respectively). The first row shows the average number and standard deviation of playcounts, per user and per artist, computed from the values of the red plots in Figures 2 and 3. The second row shows the average number of unique artists per user (second column) and the average number of unique users per artist (third column). These numbers are computed from the blue lines in the figures. The third row reveals how often, on average, users play artists they listen to (second column) and how often artists are listened to by users who listen to them at all, on average (third column). The last row is similar to the third one, but uses the median instead of the arithmetic mean to aggregate average playcounts. It shows that there exist strong outliers in the average playcount values, both per user and per artist, because the median values are much smaller than the mean values. For instance, each user listens to each of her artists on average about 21 times, but half of all users listen to each of their artists on average only 5 times or less. Therefore, there are a few users who keep on listening to their artists over and over again, while a large majority do not listen to the same artist more than a few times, on average.

Descriptors of preference and consumption behavior

The LFM-1b dataset provides a number of additional user-specific features (cf. Table 2), in particular information about temporal listening habits and music preference in terms of mainstreaminess and novelty [12]. To characterize temporal aspects, we binned the listening events of each user into weekdays and into hours of the day, and computed the share of each user's listening events over the bins. The distribution of these shares are illustrated in Figure 4 for weekdays and in Figure 5 for hours of the day. These box plots illustrate the median of the data by a horizontal red line. The lower and upper horizontal black lines of the box indicate the 25- and 75-percentiles, respectively. The horizontal black lines further above or below represent the furthest points not considered outliers, i.e., points within 1.5 times the interquartile range. Points beyond this range are depicted as blue plus signs. The red squares illustrate the arithmetic mean.

We can observe in Figure 4 that the share of listening events does not substantially differ between working days. However, during weekend (Saturday and Sunday), there is a much larger spread. A majority of people listens less during weekends than during working days (lower median). At the same time, the top 25% of active listeners consume much more music during weekends (higher 75-percentile for Saturday, and even higher for Sunday). This is obviously the result of working and leisure habits.

In Figure 5, we see that the distribution of listening events over hours of day vary more than over weekdays. It is particularly low during early morning hours (between 4 and 7h) and peaks in the afternoon and early evening (between 17 and 22h) when many people indulge in leisure time activities. While it would be interesting to investigate whether the temporal distribution of listening events varies with demographics, which we presume, such an analysis is unfortunately out of this paper's scope and left for future work.

The main statistics of the novelty and the mainstreaminess scores (both computed on time windows of one year) are given in Table 7. We can see that most users are eager to listen to new music since the average share of new artists listened to every year is approximately 50%. On the other hand, their music taste tends to be quite diverse and far away from the mainstream since the overlap between the user's distribution of listening events and the global distri-

Table 6: Statistics of the distribution of listening events among users and artists. Values after the ± sign indicate standard deviations.

	Users	**Artists**
Playcount (PC)	8,879 ± 15,962	1,824 ± 24,745
Unique artists/users	512 ± 622	105 ± 733
Mean PC per artist/user	21.21 ± 46.68	7.89 ± 17.83
Median PC per artist/user	5.16 ± 19.35	2.50 ± 2.98

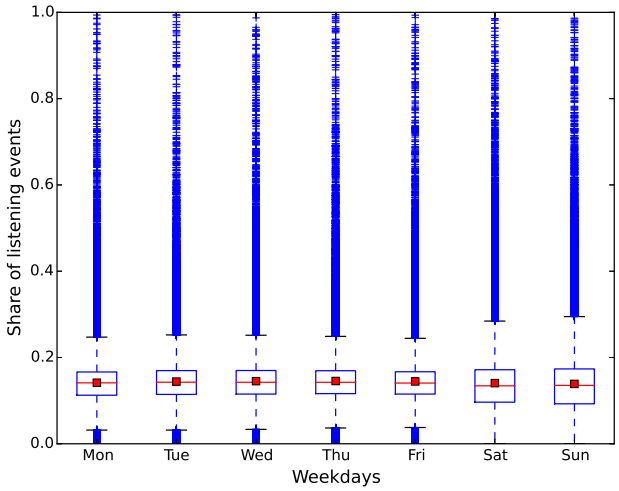

Figure 4: Distribution of listening events over weekdays.

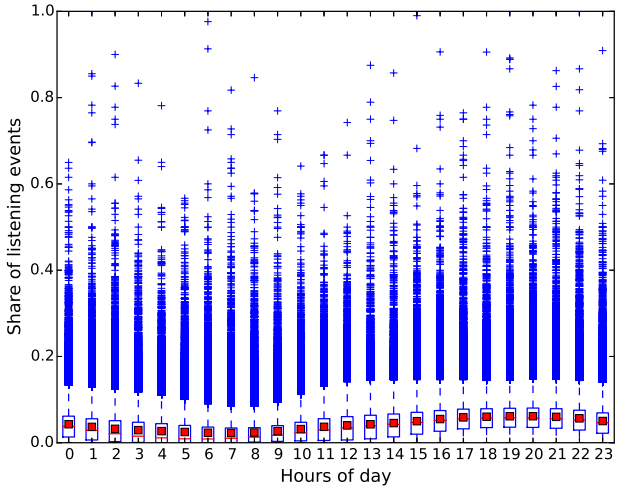

Figure 5: Distribution of listening events over hours of day. Each time range encompasses 0 to 59 minutes after the hour indicated on the x-axis.

Table 7: Statistics of novelty and mainstreaminess scores.

	Novelty	Mainstreaminess
Min.	0.000	0.000
25-perc.	0.354	0.016
Median	0.496	0.045
75-perc.	0.647	0.079
Max.	1.000	0.393
Mean	0.504	0.054
Std.	0.211	0.048

bution (that is how mainstreaminess is defined) is only 5%, on average.

3.4 Sample Source Code

To facilitate access to the dataset, we provide Python scripts that show how to load the data and perform simple computations, e.g., basic statistics, as well as how to implement a basic collaborative filtering music recommender. The code package can be found on www.cp.jku.at/datasets/LFM-1b. File `LFM-1b_stats.py` shows how to load the UAM, compute some of the statistics reported in Section 3.3, and store them in a text file. Based on this text file, `LFM-1b_plot.py` demonstrates how to create plots such as the one shown in Figures 2 and 3. In addition, we implement in `LFM-1b_recommend-CF.py` a simple memory-based collaborative filtering approach, which might serve as reference implementation and starting point for experimentation with various recommendation models.

4. MUSIC RECOMMENDATION EXPERIMENTS USING LFM-1B

Music recommendation has lately become an important task. While the LFM-1b dataset is not restricted to this task, we illustrate its use for building and evaluating a music recommender system that recommends artists. The following results are intended to serve as baseline for further experimentation and investigating more sophisticated approaches.

4.1 Recommendation Algorithms

We implemented several recommendation algorithms, detailed in the following. The results of the experiments are then presented and discussed in Section 4.2.

Collaborative Filtering

A standard memory-based collaborative filtering approach that computes the inner product of the normalized UAM (excluding the artists used for testing) was implemented. After that, the K most similar users to the target user, i.e., the user to whom we want to recommend artists, are determined and the artists these K neighbors, but not the target user, listened to are weighted with respect to their frequency among the neighbors and the similarity of each neighbor to the target. This process yields a score for each artist which is used to rank them. Finally, the top N artists are recommended. For our experiments, we set $K = 25$.

Demographic Filtering

Based on users' gender, age, and country, we define a user-user similarity matrix, from which we identify the K most similar users to the target user and eventually recommend artists using the same weighting as in the CF approach. Demographic similarity is defined binary for gender (1 if same gender, 0 otherwise), and graded for age and country (e.g., 0.8 if the age difference is between 1 and 2 years, 0.2 if the age difference is between 9 and 15 years; 1 if the users reside in the same country, 0.1 if the distance between countries — measured between their midpoint of landmass — is larger than 3,500 kilometers). We then combine these three similarity functions linearly, giving equal weights to all components. Aggregation and recommendation is performed as in the collaborative filtering approach.

Content-based Recommendation

We implemented two content-based approaches, based on different data sources. Exploiting the artist names, we fetch for each artist (i) the mood descriptors from Allmusic[15] and

[15]http://www.allmusic.com

(ii) the links present on the artist's Wikipedia[16] page.[17] We assume that artists that share moods and links are more similar. Each artist is eventually represented by a set of moods and a set of Wikipedia links, based on which two content-based recommenders are constructed. To estimate similarity between two artists, we calculate the Jaccard index between their sets of moods and between their sets of links, i.e., we compute the share of overlapping elements in both artists' item sets, separately for mood and for links. Artists similar to the ones listened to by the target user are then determined, weighted, aggregated, and ranked in a similar way than in the CF approach. Eventually, the N artists with highest scores, not known by the target user, are recommended. In our experiments, we considered up to $K = 25$ most similar artists for each artist in the target listener's training set.

Hybrid Recommender

In order to create a hybrid recommender, we follow a late fusion strategy by integrating the results of the content-based and the collaborative filtering recommenders. To this end, we first median-normalize the ranking scores given by the two recommenders to fuse. For artists suggested by both recommenders, we compute the new score as the arithmetic mean of both original scores; for all others, we take the original normalized scores. Based on the ranking obtained by sorting with respect to the new scores, we eventually recommend the top N artists.

Popularity-based Recommendation

This recommender simply sorts all artists according to their overall playcounts and recommends the top N, excluding those which the target user already knows.

Random Baselines

To contextualize the results of the recommender systems algorithms, we implemented two baselines: one that randomly selects N artists out of all artists the target user has not listened to, and one that randomly selects users and recommends N artists they listened to and are unknown to the target user.

4.2 Experiments and Results

For computational reasons, we ran the evaluation experiments on a subset of 1,100 users randomly sampled from LFM-1b. We performed 10-fold cross-validation on the listener level, i.e., we used 90% of each target user's listening history for training the system and the remaining 10% as ground truth to evaluate the recommendations made by the system. We repeated this procedure 10 times in a way that each listening event of the user occurs exactly once in the 10% test data. Varying the number of recommended artists N allows us to investigate precision at different levels of recall. The results are shown in Figure 6. As expected, CF and hybrid recommendations outperform all others. While CF has a slightly better performance when recommending a small number of artists N (higher precision at same recall), the hybrid approach outperforms CF for a large number of recommendations (higher precision and higher recall). The content-based recommender based on Wikipedia links also performs considerably well, in contrast to the mood-based one, for which data seems too sparse. All others perform substantially worse. Among the baselines, the random user selection performs slightly better than the random artist selection, which is due to the fact that the former tends to recommend artists that are more frequently listened to, while the latter performs a completely random selection.

[16]http://en.wikipedia.org

[17]To determine the correct Allmusic and Wikipedia pages for a given artist, we implemented several heuristics and filtering pipelines, a discussion of which is unfortunately not possible due to space limitations.

5. CONCLUSIONS AND FUTURE WORK

We presented the LFM-1b dataset that enables large-scale experimentation in music retrieval and recommendation. It can be downloaded from www.cp.jku.at/datasets/LFM-1b and provides information on the level of artists, albums, tracks, and users, as well as individual listening events. In addition to this standard content seen in other datasets as well, a unique feature of the LFM-1b dataset — next to its size — is the inclusion of detailed additional user-specific descriptors that model music preferences and consumption behavior. We strongly believe that this dataset, if not becoming a standard in benchmarking user-aware music recommendation approaches that go beyond rating prediction, will at least nicely complement existing datasets.

While the LFM-1b dataset can be used for experimentation in music retrieval and recommendation, particularly for collaborative filtering, demographic filtering, and personalized approaches, we contemplate several extensions. In particular, we would like to add audio-based features that allow to build content-based recommenders and retrieval systems. While audio is generally not available for the tracks in the dataset, preview snippets provided by several online music stores could be acquired and audio features computed thereon. Next to audio descriptors, features modeling the music context or background, such as TF·IDF weights computed on web pages related to artists or on lyrics, could be included too. Finally, we are investigating additional user-specific features relating to music consumption behavior, which we plan to include in a possible extension of the current dataset.

6. ACKNOWLEDGMENTS

This research is supported by the Austrian Science Fund (FWF): P25655. It would not have been possible without the help of numerous students, in particular, David Hauger, Andreas Frank, Matthias Freßdorf, Fabian Schneider, Georg Eschbacher, Hubert Scharfetter, and Nino Kratzer. Finally, we would like to thank Last.fm for providing an extensive API and for their liberal way of allowing to share data provided by their API, for non-commercial purposes. This strongly supports research in music recommendation, as evidenced by many publications that exploit Last.fm data.

7. REFERENCES

[1] G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin. Context-aware Recommender Systems. *AI Magazine*, 32:67–80, 2011.

[2] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere. The Million Song Dataset. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, Miami, FL, USA, October 2011.

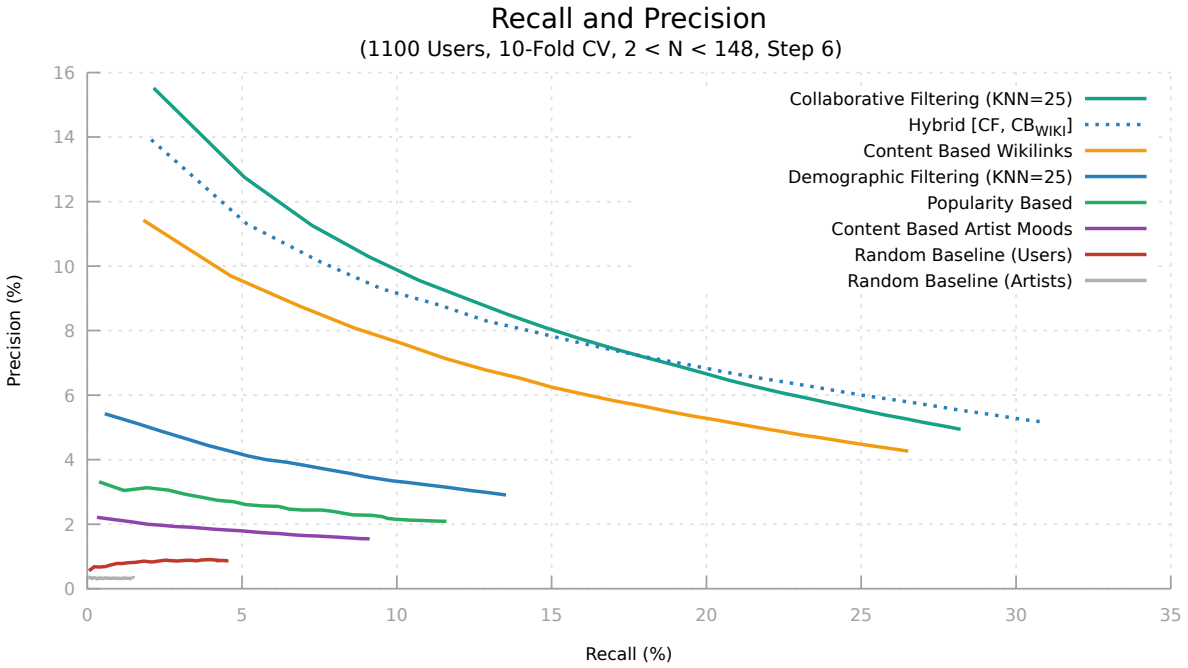

Figure 6: Precision/Recall plot of various recommendation algorithms, applied to a random subset of 1,100 users from the LFM-1b dataset. Ten-fold cross-validation was used. Precision and recall are plotted for various numbers of recommended artists N, ranging from 2 to 148 using a step size of 6.

[3] O. Celma. *Music Recommendation and Discovery – The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer, Berlin, Heidelberg, Germany, 2010.

[4] G. Dror, N. Koenigstein, Y. Koren, and M. Weimer. The Yahoo! Music Dataset and KDD-Cup'11. *JMLR: Proceedings of KDD-Cup 2011 Competition*, 18:3–18, October 2012.

[5] M. Grachten, M. Schedl, T. Pohle, and G. Widmer. The ISMIR Cloud: A Decade of ISMIR Conferences at Your Fingertips. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, Kobe, Japan, October 2009.

[6] D. Hauger, M. Schedl, A. Košir, and M. Tkalčič. The Million Musical Tweets Dataset: What Can We Learn From Microblogs. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, Curitiba, Brazil, November 2013.

[7] E. Law, K. West, M. Mandel, M. Bay, and J. S. Downie. Evaluation of Algorithms Using Games: The Case of Music Annotation. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, the Netherlands, August 2010.

[8] J. H. Lee, M. C. Jones, and J. S. Downie. An Analysis of ISMIR Proceedings: Patterns of Authorship, Topic, and Citation. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, Kobe, Japan, October 2009.

[9] C. Liem, M. Müller, D. Eck, G. Tzanetakis, and A. Hanjalic. The Need for Music Information Retrieval with User-centered and Multimodal Strategies. In *Proceedings of the 1st International ACM Workshop on Music Information Retrieval with User-centered and Multimodal Strategies (MIRUM)*, Scottsdale, AZ, USA, November 2011.

[10] B. McFee, T. Bertin-Mahieux, D. Ellis, and G. Lanckriet. The Million Song Dataset Challenge. In *Proceedings of the 4th International Workshop on Advances in Music Information Research (AdMIRe)*, Lyon, France, April 2012.

[11] M. Müller, M. Goto, and M. Schedl, editors. *Multimodal Music Processing*, volume 3 of *Dagstuhl Follow-Ups*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2012.

[12] M. Schedl and D. Hauger. Tailoring Music Recommendations to Users by Considering Diversity, Mainstreaminess, and Novelty. In *Proceedings of the 38th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Santiago, Chile, August 2015.

[13] M. Schedl, P. Knees, B. McFee, D. Bogdanov, and M. Kaminskas. *Recommender Systems Handbook*, chapter Music Recommender Systems. Springer, 2nd edition, 2015.

[14] B. L. Sturm. The State of the Art Ten Years After a State of the Art: Future Research in Music Information Retrieval. *Journal of New Music Research*, 43(2):147–172, 2014.

[15] Yuan Cao Zhang, Diarmuid O Seaghdha, Daniele Quercia, Tamas Jambor. Auralist: Introducing Serendipity into Music Recommendation. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM)*, Seattle, WA, USA, February 2012.