The Million Song Dataset Challenge

Brian McFee*
CAL Lab
UC San Diego
San Diego, USA
bmcfee@cs.ucsd.edu

Thierry Bertin-Mahieux*
LabROSA
Columbia University
New York, USA
tb2332@columbia.edu

Daniel P.W. Ellis
LabROSA
Columbia University
New York, USA
dpwe@ee.columbia.edu

Gert R.G. Lanckriet
CAL Lab
UC San Diego
San Diego, USA
gert@ece.ucsd.edu

ABSTRACT

We introduce the Million Song Dataset Challenge: a large-scale, personalized music recommendation challenge, where the goal is to predict the songs that a user will listen to, given both the user's listening history and full information (including meta-data and content analysis) for all songs. We explain the taste profile data, our goals and design choices in creating the challenge, and present baseline results using simple, off-the-shelf recommendation algorithms.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Music recommendation

Keywords

Music information retrieval, recommender systems

1. INTRODUCTION

With the rise of digital content distribution, people now have access to music collections on an unprecedented scale. Commercial music libraries easily exceed 15 million songs, which vastly exceeds the listening capability of any single person; even one million songs would take more than seven years of non-stop listening. In order to help users cope with the rapidly expanding catalogue of readily available music, numerous academic efforts have been put forward to automate search, annotation and retrieval of musical content [2, 6, 22, 29]. Clearly, the advancement of *music information retrieval* (MIR) technology can have far-reaching implications for commercial music recommendation systems, such as Amazon,[1] Google,[2] Last.fm,[3] Pandora,[4] and Spotify.[5] However, at present, the vast majority of academic efforts ignore user-centric music recommendation, and instead focus on the simpler, related tasks of automatic semantic annotation and similarity evaluation.

The primary reason for this lack of recommender focus is immediately obvious to researchers: the lack of publicly available, open and transparent data for personalized recommendation has prevented academic research on the problem. Developers of a commercial music recommender system are of course free to build and evaluate models using the system's user base, but privacy and intellectual property concerns typically prevent the open evaluation and publication of data which are crucial to academic study. While there has been some collaboration between industry and academics to provide open data for music recommendation [10], the resulting data typically consists of a bare minimum of per-song meta-data, making it difficult to apply content-based methods to a standard reference data set.

To help open the door to reproducible, open evaluation of user-centric music recommendation algorithms, we have developed the Million Song Dataset Challenge. Our goals in the design of this contest are twofold. First, and most importantly, the data is *open*: meta-data, audio content-analysis, and standardized identifiers are available for all songs. Second, the data is *large-scale*, and comprised of listening histories from over one million users, providing a realistic simulation of industrial recommendation settings. We hope that the release of this data and ensuing competition will facilitate a long line of academic research into the domain-specific aspects of music recommendation, from which we may gain a clearer understanding of how the problem differs from the more commonly studied recommendation domains (*e.g.*, movies or books).

WWW 2012 Companion, April 16–20, 2012, Lyon, France.
ACM 978-1-4503-1230-1/12/04.

* The first two authors contributed equally to this paper.

[1] http://www.amazon.com/music
[2] http://www.google.com/music
[3] http://www.last.fm
[4] http://www.pandora.com
[5] http://www.spotify.com

2. RELATED EVALUATIONS

By far, the most famous example of a large-scale recommendation dataset is the Netflix challenge [5]. The Netflix challenge data consists of approximately 100 million time-stamped, explicit ratings (1 to 5 stars) of over 17,000 movies by roughly 480,000 users. The data was released in the form of a competition, in which different teams competed to reduce the root mean-squared error (RMSE) when predicting withheld ratings.

In the music domain, the recent 2011 KDD Cup followed the basic structure of the Netflix challenge, but was designed around data provided by Yahoo! Music[6] [14]. Like Netflix, the KDD Cup data consists of explicit ratings (on a scale of 0–100) of items by users, where the item set is a mixture of songs, albums, artists, and genres. The KDD-Cup'11 set is quite large-scale — over 260 million ratings, 620,000 items and one million users — but is also anonymous: both the users and items are represented by opaque identifiers. This anonymity makes it impossible to integrate meta-data (beyond the provided artist/genre/album/track taxonomy) and content analysis when forming recommendations. While the KDD Cup data is useful as a benchmark for content-agnostic collaborative filtering algorithms, it does not realistically simulate the challenges faced by commercial music recommendation systems.

Within the content-based music information retrieval domain, previous competitions and evaluation exchanges have not specifically addressed user-centric recommendation. Since 2005, the annual Music Information Retrieval Evaluation eXchange (MIREX) has provided a common evaluation framework for researchers to test and compare their algorithms on standardized datasets [13]. The majority of MIREX tasks involve the extraction and summarization of song-level characteristics, such as genre prediction, melody extraction, and artist identification. Similarly, the MusiCLEF evaluation focuses on automatic categorization, rather than directly evaluating recommendation [26].

Within MIREX, the most closely related tasks to recommendation are audio- and symbolic-music similarity (AMS, SMS), in which algorithms must predict the similarity between pairs of songs, and human judges evaluate the quality of these predictions. Also, as has been noted often in the MIR literature, similarity is not synonymous with recommendation, primarily because there is no notion of *personalization* [11]. Moreover, the MIREX evaluations are not *open* in the sense that the data is not made available to researchers. Because meta-data (*e.g.*, artist identifiers, tags, etc) is not available, it is impossible to integrate the wide variety of readily available linked data when determining similarity. Finally, because the evaluation relies upon a small number of human judges, the size of the collection is necessarily limited to cover no more than a few thousand songs, orders of magnitude smaller than the libraries of even the smallest commercially viable music services. Thus, while the current MIREX evaluations provide a valuable academic service, they do not adequately simulate realistic recommendation settings.

We note that previous research efforts have attempted to quantify music similarity via human judges [15]. Most of the issues mentioned above regarding MIREX also apply, in particular the small number of human judges.

[6] http://music.yahoo.com

3. AN OPEN DATA MODEL

As noted in Section 2, previous music recommendation studies and contests were either done on a small scale (few songs, few users), or reduced to a pure collaborative filtering problem (*i.e.*, by anonymizing the artists). Our goal is to provide a large-scale, transparent music recommendation challenge, which will allow participants to exploit both meta-data and content analysis.

3.1 Song information

The core of any music collection data is the meta-data which lets you identify the song. Typically, the artist name and song title is sufficient as a starting point. From such a list of artists and titles, researchers could in theory gather all information pertinent to these songs, including audio and human judgements, and use that to make recommendations.

Obviously, finding all pertinent information is a tremendous task, thus we provide a few significant shortcuts. As indicated by its name, the challenge is organized using songs in the Million Song Dataset (MSD): a freely-available collection of audio features and meta-data for a million contemporary popular music tracks [7]. Comprising several complementary datasets that are linked to the same set of songs, the MSD contains extensive meta-data, audio features, tags on the artist- and song-level, lyrics, cover songs, similar artists, and similar songs [8]. Due to copyright restrictions, the audio cannot be distributed, but snippets can be fetched from 7digital[7] and streamed from Rdio.[8] The meta-data also makes it easy to link tracks and artists to online resources and application programming interfaces (APIs), such as those provided by The Echo Nest, MusicBrainz, Last.fm, musiXmatch, and Musicmetric. Using these resources, one can add to the MSD biographical information, online popularity measures, album covers, tour dates, twitter feeds, lists of awards and many more, all possibly relevant to music recommendation. User data from other sources could even be added, for instance, the MusicBrainz identifiers enable matching against the "Last.fm dataset 360K" [10].[9]

Transparency of the dataset could also enable a variety of less traditional approaches to music recommendation. For example, it would be possible — albeit expensive or possibly inefficient — to crowd-source the prediction task by providing audio clips for a user's taste profile to a pool of human expert (or amateur) recommenders. The idea of using humans on that scale may seem far-fetched, but Pandora is a successful commercial system based on human annotations [31]. Similarly, games designed to collect tags [20, 23, 32] could also be modified in order to collect recommendations. Ideas from the DARPA Network Challenge could also be adapted, *i.e.*, using members of social networks to solve a task [27].

3.2 Taste profiles

The collection of data we use is known as the Taste Profile Subset.[10] It consists of more than 48 million triplets

[7] http://us.7digital.com/

[8] http://labrosa.ee.columbia.edu/millionsong/blog/11-10-31-rdio-ids-more-soon

[9] http://mtg.upf.edu/node/1671

[10] http://labrosa.ee.columbia.edu/millionsong/tasteprofile

```
b80344d063b5ccb3...  SOYHEPA12A8C13097F  8
b80344d063b5ccb3...  SOYYWMD12A58A7BCC9  1
b80344d063b5ccb3...  SOZGCUB12A8C133997  1
b80344d063b5ccb3...  SOZOBWN12A8C130999  1
b80344d063b5ccb3...  SOZZHXI12A8C13BF7D  1
85c1f87fea955d09...  SOACWYB12AF729E581  2
85c1f87fea955d09...  SOAUSXX12A8C136188  1
85c1f87fea955d09...  SOBVAHM12A8C13C4CB  1
85c1f87fea955d09...  SODJTHN12AF72A8FCD  2
```

Figure 1: A few lines of the raw data of the Taste Profile Subset. The three columns are *user ID*, *song ID* and *play count*. The *user ID*s have been truncated for visualization purposes.

	Netflix	KDD-Cup'11	MSD
# Items	17K	507K	380K
# Users	480K	1M	1.2M
# Observations	100M	123M	48M
% Density	1.17%	0.02%	0.01%

Table 1: Scale and density comparison of the Netflix challenge, KDD-Cup'11 (track 1, songs only), and MSD Challenge datasets. *Density* is the fraction of observed (non-zero) entries in the users-by-items feedback matrix, *i.e.*, the portion of observed ratings (Netflix/KDD) or user-item interactions (MSD).

(*user*, *song*, *count*) gathered from user listening histories. The data was provided by an undisclosed set of applications, where each user could select the song they wanted to listen to. The data consists of approximately 1.2 million users, and covers more than 380,000 songs in MSD. A raw sample of the data is shown in Figure 1.

Due to user privacy concerns, users must remain anonymous, and we are therefore unable to provide demographic information. We are also unable to provide timestamps for listening events, which although highly informative for recommendation purposes [4, 19], may facilitate the use of de-anonymization techniques [24]. The release of the play-counts could already be considered a risk, though we believe it is limited and reasonable.

Table 1 illustrates the scale of the Taste Profile Subset in comparison to the Netflix Challenge and KDD Cup (track 1) data (including tracks, but not artists, genres or albums). However, the resulting density of observations in the user-item interaction matrix is of roughly the same order of magnitude, and both are significantly lower than in Netflix.

Figure 2 illustrates the distributions of users-per-song and songs-per-user in the Taste Profile Subset. Most importantly, there is a dramatic long-tail effect in song popularity: half of the observed songs have at most 13 unique listeners, while 95% have at most 488 unique listeners. Because the overwhelming majority of songs have so few listeners, we anticipate that the collaborative filter alone may not provide enough information to form good recommendations, and side-information (*i.e.*, content and meta-data) will form an integral component of successful recommenders. From the user perspective, however, there is significantly more information on average: all users have at least 10 songs, and roughly 50% have 30 or more.

3.3 Implicit feedback

Note that unlike the KDD Cup [14], we use play counts instead of explicit ratings. From a collaborative filtering perspective, it means we use *implicit feedback* [25]. An obvious problem is that having played a track doesn't mean it was liked [17]. However, although the user may not have had a positive reaction to a song upon listening to it, the fact that it was listened to suggests that there was at least some initial interest on behalf of the user.

Implicit feedback is also easier to gather, since it does not require the user to do anything more than listen to music, and it still provides meaningful information that can be used on its own or supplement explicit feedback methods [21]. More fundamentally, using implicit feedback from play counts makes the data (and methods developed for recommendation) more universal in the sense that any music recommendation service can store and access play counts. This contrasts with explicit feedback, where different services may provide different feedback mechanisms — *e.g.*, Pandora's *thumbs-up/thumbs-down*, Last.fm's *loved tracks*, or Yahoo! Music's explicit ratings — each with different semantics, and thus requiring different recommendation algorithms.

4. RECOMMENDER EVALUATION

In this section, we describe the evaluation metrics used in our benchmarks, and the motivation for these specific choices. We note that this is not a comprehensive list, as additional metrics will be included in the competition (although not for scoring purposes), in order to provide a more comprehensive view of the relative merits of each competing method.

4.1 Prediction: ranking

As mentioned above, previous recommendation datasets such as Netflix and KDD Cup consisted of explicit feedback from users, where each observation is a bounded numeric score quantifying the affinity (positive or negative) of the user for the item in question. In the explicit feedback regime, a perfectly sensible recommendation strategy is to predict the rating which a user would supply in response to an item, and present items in descending order according to the predicted rating. This motivates the use of root mean-squared error (RMSE) as a proxy evaluation for recommender quality.

Since we are working with implicit feedback (play-counts), which is generally non-negative and unbounded, the case for evaluation by RMSE is less clear. Instead, we will evaluate recommendation algorithms by recall of positively-associated items for each user. Specifically, for each user, the recommender observes a subset of the songs consumed by the user, and predicts a ranking over all other songs in the dataset. Ideally, the remaining songs consumed by the user would be ranked ahead of all other songs. Note that during evaluation, all that matters is whether the user listened to the song or not, and the exact number of times the song was consumed is not important. This information is of course available for training observations, and it is up to the discretion of the algorithm designer how best to use it.

In the live competition, teams will be required to upload their predictions to a central server, which will then perform the evaluation on withheld test data, and provide a score and leader-board. Due to the large scale of the test data

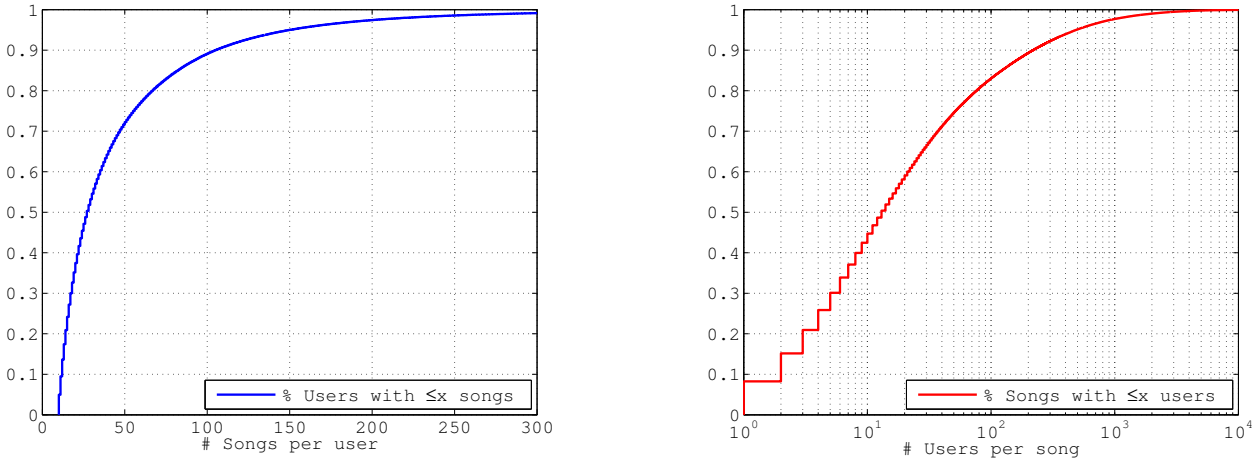

Figure 2: Left: the empirical cumulative distribution of users as a function of library size. Right: the cumulative distribution of songs as a function of number of unique users. Statistics were estimated from the training sample of 1 million users.

(100K users and 380K songs), transferring and storing a full ranking of all songs for each test user would require an unreasonably large amount of bandwidth and storage. Instead, we will set a threshold τ and only evaluate the top-τ portion of the predicted rankings.

The choice of the cutoff τ will depend on the minimum quality of the recommenders being evaluated, which we attempt to assess Section 5. For instance, if we only consider the top $\tau = 100$ predicted songs for each user, and none of the actual play counts are covered by those 100 songs, all recommenders will be considered equally disappointing. The baselines investigated in Section 5 suggest a reasonable range for τ. In particular, the popularity baseline gives a reasonable lower bound on a recommender's performance at any given cutoff point, see Figure 3. For the remainder of this paper, we fix $\tau = 10^4$, but it may be lowered in the final contest.

We acknowledge that there are aspects that can not be measured from the play counts and a ranked list of predictions. Specifically, ranked-list evaluation — indeed, virtually any conceivable off-line evaluation — may not sufficiently capture the human reaction: novelty and serendipity of recommendations, overall listening time, or changes in the user's mood. The only real solution to this "human factor" problem would be A-B testing [30] on an actual commercial system, such as Last.fm or Pandora. Clearly, live A-B testing would not constitute a practical evaluation method for a public competition, so we must make do with off-line evaluations.

4.2 Truncated mAP

In order to define the evaluation metrics, we must first define some notation. Let $M \in \{0,1\}^{m \times n}$ denote the binarized users-by-items feedback matrix, where $M_{u,i}$ is 1 if user u listened to item i, and 0 otherwise. Let y denote a ranking over items, where $y(j) = i$ indicates that item i is ranked at position j. It will be assumed that a predicted ranking y_u for a user u omits the items already known to be played by the user (*i.e.*, the training feedback).

The primary evaluation metric is mean average precision (mAP) of the truncated ranking. The mAP metric emphasizes the top recommendations, and is commonly used throughout the information retrieval literature [3]. For any $k \leq \tau$, the precision-at-k (P_k) is the proportion of correct recommendations within the top-k of the predicted ranking:

$$P_k(u, y) = \frac{1}{k} \sum_{j=1}^{k} M_{u,y(j)}. \tag{1}$$

For each user, we now take the average precision at each recall point:

$$\mathrm{AP}(u, y) = \frac{1}{n_u} \sum_{k=1}^{\tau} P_k(u, y) \cdot M_{u,y(k)}, \tag{2}$$

where n_u is the smaller of τ and the number of positively associated songs for user u. Finally, averaging over all m users, we have the mean average precision:

$$\mathrm{mAP} = \frac{1}{m} \sum_{u} \mathrm{AP}(u, y_u), \tag{3}$$

where y_u is the ranking predicted for user u.

4.3 Additional metrics

In order to present a more holistic view of each baseline recommender's performance, we report a variety of standard information retrieval metrics in addition to the primary score (meanAP). Specifically, we include the mean reciprocal rank (MRR) [33], normalized discounted cumulative gain (nDCG) [18], precision-at-10 (P_{10}), and recall at the cutoff point R_τ.

MRR rewards a ranking y for placing the first relevant result early in the list (regardless of the positions of remaining relevant results), and is often used to evaluate performance of search engines for factual queries (*i.e.*, where only one good result is needed). In the context of music recommendation, this is roughly analogous to the setting where a user simply wants to find a song to play immediately, and stops looking once a satisfactory recommendation is found.

nDCG is similar to mAP, in that predictions early in the ranking are weighted higher than those later in the ranking. Each relevant result i receives a reward discounted by its position $1/\log(y^{-1}(i))$, and the sum over all results is normalized according to the score of the ideal ranking.

Precision-at-10 can be interpreted as the mean accuracy of the first page/screen of recommendations. We include P_{10} as a proxy evaluation for a user's first impression of the recommender.

Finally, recall-at-τ computes the fraction of the user's library retrieved within the truncated ranking.

We note that this list does not capture all possible aspects of a recommender, and additional metrics may be included in the actual contest.

5. BASELINE EXPERIMENTS

In this section, we describe the three algorithms used to produce baseline results: a global popularity-based recommender with no personalization, a simple recommender which predicts songs by artists already present in the user's taste profile, and finally a latent factor model.

All results are based on a train-test split that is similar, but may differ from the split to be used in the contest. The training set consists of the full taste profiles for $\sim$ 1M users, and partial taste profiles for the 10K test users. We report results on 10K test users, which would be the size of the validation set in the actual contest, as opposed to 100K users for the official results.

5.1 Popularity - song bias

The most trivial recommendation algorithm is to simply present each song in descending order of its popularity, regardless of the user's taste profile. Although this model does not incorporate any personalization, it is useful to establish a global baseline that should be out-performed by any reasonable personalized model. The relative differences in performance between this global model and personalized models also provide insight into the predictability of users, and the influence of songs in the long tail.

Concretely, if M denotes the user-song interaction matrix (populated only with training data), then each song i receives a weight

$$w_i = \sum_u M_{u,i}. \tag{4}$$

At test time, for a user u, songs are ranked in descending order of w_i, skipping those songs already consumed by the user.

5.2 Same artist - greatest hits

As a second baseline, we constructed a recommender which simply produces the most popular songs — the *greatest hits* — by artists that the user has already listened to. Concretely, we first compute the score w_i for each song is computed by (4), and the ranking is re-ordered to place songs by artists in the user's profile first (ordered by popularity), followed by the remaining songs (also ordered by popularity).

The greatest hits recommender provides a bare minimum level of personalization above the global popularity model described above. This model is also maximally conservative, in the sense that it does not explore the space beyond songs with which the user is likely already familiar. Although this model on its own would not produce a satisfying user experience, it has been shown in empirical studies that including these conservative or obvious recommendations can help build the user's trust in the system. We also note that without transparent item-level meta-data, such a recommender would not be possible.

5.3 Latent factor model - BPR

Latent factor models are among the most successful recommender system algorithms [9, 12]. In its simplest form, a latent factor model decomposes the feedback matrix M into a latent feature space which relates users to items (and vice versa). This is usually achieved by factoring the matrix M as

$$M \approx \widehat{M} = U^{\mathsf{T}} V,$$

where $U \in \mathbb{R}^{k \times m}$ is a low-rank representation of the m users, and $V \in \mathbb{R}^{k \times n}$ represents the n items. Once the representations U and V have been constructed, personalized recommendation is performed for a user u by ranking each item i in descending order of the predicted feedback

$$w_i = \widehat{M}_{u,i} = U_u^{\mathsf{T}} V_i.$$

In recent years, various formulations of this general approach have been developed to solve collaborative filter recommendation problems under different assumptions. For our baseline experiments, we applied the Bayesian personalized ranking matrix factorization (BPR-MF) model [28]. To facilitate reproducibility, we use the open source implementation provided with the MyMediaLite software package [16].

The BPR-MF model is specifically designed for the implicit feedback (positive-only) setting. At a high level, the algorithm learns U and V such that for each user, the positive (consumed) items are ranked higher than the negative (no-feedback items). This is accomplished by performing stochastic gradient descent on the following objective function:

$$f(U,V) = \sum_{\substack{u,i,j: \\ M_{u,i} > M_{u,j}}} \ln\left(1 + e^{-U_u^{\mathsf{T}}(V_i - V_j)}\right) + \lambda_1 \|U\|_{\mathsf{F}}^2 + \lambda_2 \|V\|_{\mathsf{F}}^2,$$

where u is a user, and i and j are positive and negative items for user u. The first term is the logistic loss for incorrectly ranking item j ahead of item u. The remaining terms regularize the solution to prevent the model from overfitting, and are balanced with the loss function by parameters $\lambda_1, \lambda_2 > 0$.

For our baseline experiments, we fix the number of latent factors to $k = 32$, and vary $\lambda_1 = \lambda_2 \in \{10^{-5}, 10^{-4}, \ldots, 1\}$. We report the scores for $\lambda_1 = \lambda_2 = 10^{-4}$, which achieved the highest meanAP on the validation set.

5.4 Baseline results

The results of our baseline experiments are presented in Table 2.

From these results, we observe that the top 10K most popular songs account for nearly 60% of an average test user's play counts. In other words, more than half of user-song interactions occur within the top 3% of songs, indicating a significant bias toward massively popular items. This observation supports two frequently noted aspects of recommender systems problems: 1) popularity (item-bias) must be taken into account when making recommendations, and

	mAP	MRR	nDCG	P_{10}	R_τ
Popularity	0.026	0.128	0.177	0.046	0.583
Same artist	**0.095**	**0.323**	**0.297**	**0.136**	0.731
BPR-MF	0.031	0.134	0.210	0.043	**0.733**

Table 2: Baseline results for the three methods described in Section 5. Rankings are evaluated with a cutoff of $\tau = 10^4$. In all cases, higher is better (1 ideal, 0 awful).

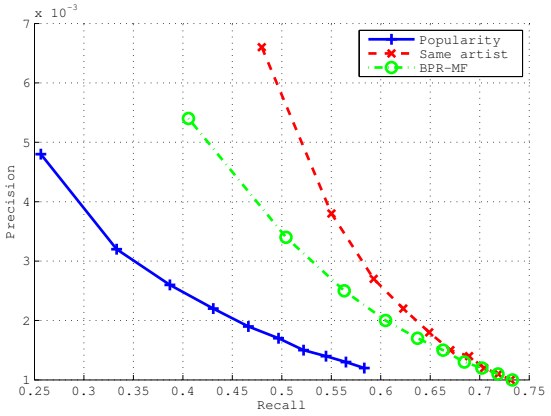

Figure 3: Precision-recall curves for each baseline algorithm. Rankings are evaluated at cutoffs $\{1000, 2000, \ldots, 10000\}$.

2) due to sparsity of observations, a pure collaborative filter model may have difficulty forming recommendations in the long tail.

We also observe that the same-artist recommender significantly outperforms the global popularity model, as well as the latent factor model. This indicates that there are substantial gains to be had by exploiting meta-data transparency, and incorporating domain knowledge specific to the structure of musical items, *e.g.*, the relationship between songs, artists and albums. A similar result was also observed in the KDD-Cup'11 data, where the provided item taxonomy was used to significantly improve performance [19]. The key difference here is that there is no single taxonomy in MSD; rather, data transparency will allow practitioners to experiment with different taxonomies and additional side-information in order to produce the best possible recommendations.

Figure 3 illustrates the precision-recall curves for each baseline recommender, which follow the same relative ordering exhibited by the metrics in Table 2. We note that the BPR-MF and same-artist recommenders converge to comparable precision-recall points for large cut-offs, and only differ significantly early in the ranked list.

6. DISCUSSION

Obviously, a valid reason for not holding this evaluation until now is because we could not, especially from a data point of view. Beyond that, there is a current trend towards organizing contests, but we did not prepare this challenge simply to follow the crowd. We believe that releasing the taste profile data in the form of a research challenge can be beneficial for many groups of researchers.

6.1 Music Information Retrieval

The MIR communityhas long believed that music is not a simple product or commodity that can be managed without the benefit of specific musical domain knowledge. In addition to audio content, cultural context (*e.g.*, artist, release year, language, and other meta-data) around a song should be relevant, especially when operating in the long-tail/cold-start regimes that frequently hinder traditional collaborative filter methods [10]. This stands in stark contrast to the fact that most commercial recommenders (such as iTunes and Amazon) appear to rely more upon collaborative filtering techniques than content analysis. However, there has yet to be a sufficiently realistic, head-to-head evaluation comparing content-based and collaborative-filter music recommendation, so it is currently unclear how much content analysis can help (if at all).

By encouraging submissions from the recommender systems community (as well as MIR), we hope that the MSD Challenge will provide some hard data to guide future research in content-based recommendation. This challenge also provides an opportunity for many of the sub-branches of the field to work together. No one knows for sure which methods will generate the best recommendations, and teams using content-based, online interactions, human judgment and musicology all have a fair shot at winning. Given the historical trends in recommendation challenges (Netflix and KDD Cup), we expect that successful teams will employ hybrid models which integrate many different sources of information, thus encouraging collaboration between diverse groups of researchers.

6.2 Open data mining

Looking beyond music recommendation to more general multi-media domains (*e.g.*, YouTube videos), we emphasize the need for open access to content, as well as meta-data. Demonstrating the utility of content-based techniques in recommendation is the best way to convince large-scale service providers to allow researchers access to their content. To be fair, there has been industrial interest in content-based multimedia management [1, 34] and there are some audio sharing systems being tested for developers, *e.g.*, EMI and The Echo Nest[11], but researchers still have a hard time accessing audio content.

By providing open data, we also stand to attract researchers from other fields (*e.g.*, machine learning and data mining) and raise awareness of MIR problems in general. The MSD Challenge provides a large, high-dimensional, richly annotated data set, which may be of general interest to machine learning researchers. Strengthening the ties between the MIR, machine learning, and data mining communities can only benefit all parties involved, and open the door to future collaborations.

6.3 Industrial applications

Finally, developers at music start-ups would greatly benefit from a thorough evaluation of music recommendation

[11] `http://developer.echonest.com/sandbox/emi/`

systems. Although not the primary goal of the MSD Challenge, we hope that the results of the competition will influence and improve the overall quality of practical music recommendation technology for all users, not just MIR researchers.

6.4 Challenge details

At the time of writing, we hope to release the data by March, have the contest run until August, and announce winners in September. The main contest would be hosted by Kaggle,[12] and we have ongoing discussions with MIREX, ISMIR and WOMRAD to present the findings. We are also pursuing the possibility of prizes for the winner(s), provided by various corporations. The Kaggle leader-board would be based on a validation set of 10K users, and the final results would use a different set of 100K users. The training data would contain 1M users, and we would set aside another 110K users to run a second edition. The second edition might be "netflix-style": it could go on until a certain improvement has been achieved relative to a solid baseline. After the first edition (and the second if it happens), the test data would be released so that future systems can be tested and compared on the same set.

7. CONCLUSION

We have presented the MSD Challenge: our attempt at making the best offline evaluation (no A-B testing possible) with the restriction of guaranteeing the anonymity of the listeners. The scale and transparency of the data makes it unlike any other previous contest. This is an opportunity for the MIR field to show its strength on an industrial-size challenge and to merge the results from different sub-fields into one system for one specific task. We also hope that this contest will better expose MIR techniques and applications to other researchers across other fields. This work's goal is to clearly describe our motivations and implementation decisions. Knowing the challenge's goals, strengths and limitations is essential for the contest to be successful.

8. ACKNOWLEDGMENTS

The authors would like to thank Paul Lamere and Brian Whitman of The Echo Nest for providing the taste profile data. Additionally, we thank the MSD Challenge steering committee, and the anonymous reviewers for their helpful comments and suggestions.

This work was supported in part by grants IIS-0713334, IIS-1117015, CCF-0830535, and IIS-1054960 from the National Science Foundation, and by gifts from Google, Inc. T.B.M. is supported in part by a NSERC scholarship. B.M. and G.R.G.L. further acknowledge support from Qualcomm, Inc., Yahoo!, Inc., the Hellman Fellowship Program, and the Sloan Foundation. This research was supported in part by the UCSD FWGrid Project, NSF Research Infrastructure Grant Number EIA-0303622.

9. REFERENCES

[1] H. Aradhye, G. Toderici, and J. Yagnik. Video2text: Learning to annotate video content. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*, pages 144–151. IEEE, 2009.

[2] J.-J. Aucouturier and F. Pachet. Music similarity measures: What's the use. In Michael Fingerhut, editor, *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, pages 31–38, Oct 2002.

[3] Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[4] L. Baltrunas and X. Amatriain. Towards time-dependant recommendation based on implicit feedback. In *Workshop on Context-aware Recommender Systems (CARS'09)*, 2009.

[5] J. Bennett and S. Lanning. The netflix prize. In *Proceedings of KDD Cup and Workshop*, volume 2007, page 35, 2007.

[6] A. Berenzweig, B. Logan, D. Ellis, and B. Whitman. A large-scale evaluation of acoustic and subjective music similarity measures. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)*, 2003.

[7] T. Bertin-Mahieux, D. Ellis, B. Whitman, and P. Lamere. The million song dataset. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011.

[8] T. Bertin-Mahieux and D.P.W. Ellis. Large-scale cover song recognition using hashed chroma landmarks. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Platz, NY, 2011.

[9] D. Billsus and M.J. Pazzani. Learning collaborative information filters. In *Proceedings of the Fifteenth International Conference on Machine Learning*, volume 54, page 48, 1998.

[10] Ò. Celma. *Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space*. Springer-Verlag New York Inc, 2010.

[11] Òscar Celma and Paul Lamere. Music recommendation and discovery revisited, 2011. ACM Conference on Recommender Systems (Tutorial).

[12] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[13] J.S. Downie. The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoustical Science and Technology*, 29(4):247–255, 2008.

[14] G. Dror, N. Koenigstein, Y. Koren, and M. Weimer. The yahoo! music dataset and kdd-cup '11. In *KDD-Cup Workshop*, 2011.

[15] D. Ellis, B. Whitman, A. Berenzweig, and S. Lawrence. The quest for ground truth in musical artist similarity. In *Proceedings of the 3th International Conference on Music Information Retrieval (ISMIR 2002)*, 2002.

[16] Zeno Gantner, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. MyMediaLite: A free recommender system library. In *Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011)*, 2011.

[17] Y. Hu, Y. Koren, and C. Volinsky. Collaborative

[12] `http://www.kaggle.com`

filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. IEEE, 2008.

[18] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20:422–446, October 2002.

[19] Noam Koenigstein, Gideon Dror, and Yehuda Koren. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the fifth ACM conference on Recommender systems*, RecSys '11, pages 165–172, 2011.

[20] E. Law, L. von Ahn, R. Dannenberg, and M. Crawford. Tagatune: a game for music and sound annotation. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, 2007.

[21] Nathan N. Liu, Evan W. Xiang, Min Zhao, and Qiang Yang. Unifying explicit and implicit feedback for collaborative filtering. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1445–1448, New York, NY, USA, 2010. ACM.

[22] B. Logan and A. Salomon. A music similarity function based on signal analysis. In *ICME 2001*, 2001.

[23] M. Mandel and D. Ellis. A web-based game for collecting music metadata. *Journal of New Music Research, special issue: "From genres to tags: Music Information Retrieval in the era of folksonomies."*, 37(2):151–165, June 2008.

[24] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 111–125. IEEE, 2008.

[25] D.W. Oard and J. Kim. Implicit feedback for recommender systems. In *Proceedings of the AAAI Workshop on Recommender Systems*, pages 81–83, 1998.

[26] N. Orio, D. Rizo, R. Miotto, N. Montecchio, M. Schedl, and O. Lartillot. Musiclef: A benchmark activity in multimodal music information retrieval. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011.

[27] G. Pickard, I. Rahwan, W. Pan, M. Cebrian, R. Crane, A. Madan, and A. Pentland. Time critical social mobilization: The darpa network challenge winning strategy. *Arxiv preprint arXiv:1008.3172*, 2010.

[28] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.

[29] U. Shardanand. *Social information filtering for music recommendation*. PhD thesis, Massachusetts Institute of Technology, 1994.

[30] T. Tullis and W. Albert. *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Morgan Kaufmann, 2008.

[31] D. Turnbull, L. Barrington, and G. Lanckriet. Five approaches to collecting tags for music. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR 2008)*, 2008.

[32] D. Turnbull, R. Liu, L. Barrington, and G. Lanckriet. A game-based approach for collecting semantic annotations of music. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, 2007.

[33] Ellen M. Voorhees. The trec-8 question answering track report. In *In Proceedings of TREC-8*, pages 77–82, 1999.

[34] J. Weston, S. Bengio, and P. Hamel. Large-scale music annotation and retrieval: Learning to rank in joint semantic spaces. *Arxiv preprint arXiv:1105.5196*, 2011.