CONTENT-BASED USER MODELS: MODELING THE MANY FACES OF MUSICAL PREFERENCE

Eva Zangerle
Universität Innsbruck
Department of Computer Science
eva.zangerle@uibk.ac.at

Martin Pichl
Universität Innsbruck
Department of Computer Science
martin.pichl@uibk.ac.at

ABSTRACT

User models that capture the musical preferences of users are central for many tasks in music information retrieval and music recommendation, yet, it has not been fully explored and exploited. To this end, the musical preferences of users in the context of music recommender systems have mostly been captured in collaborative filtering-based approaches. Alternatively, users can be characterized by their average listening behavior and hence, by the mean values of a set of content descriptors of tracks the users listened to. However, a user may listen to highly different tracks and genres. Thus, computing the average of all tracks does not capture the user's listening behavior well. We argue that each user may have many different preferences that depend on contextual aspects (e.g., listening to classical music when working and hard rock when doing sports) and that user models should account for these different sets of preferences. In this paper, we provide a detailed analysis and evaluation of different user models that describe a user's musical preferences based on acoustic features of tracks the user has listened to.

1. INTRODUCTION

In the last decade, the amount of tracks available on streaming platforms has literally exploded. Users are supported in exploring and wading through these music collections by means of personalization—mostly by recommender systems that provide users with a list of tracks they might like to listen to. Such personalization is central for the success of streaming platforms as it eases the task of discovering new and enjoyable music for users.

For music information retrieval (MIR) and particularly, for personalization tasks in this context, modeling the musical preferences of users is naturally a central aspect. Yet, user modeling for MIR and music recommender systems (MRS) has hardly been investigated [4,32,33]. To this end, music recommender systems have mostly been realized by means of collaborative filtering (CF) methods [16] or more advanced factorization approaches [17], where recommendations are based on interactions between users and items. Such systems are agnostic to content features as recommendations are computed based on the similarity of users (or items) based on their co-occurrence in the listening histories of all users. On the other hand, (the less adopted) content-based recommender systems [22] compute recommendations based on the similarity of content descriptors of tracks. Also, hybrid recommender systems combining CF- and content-based approaches have been proposed [7].

In the field of MIR, tracks are traditionally characterized by content descriptors—these range from detailed features such as MFCCs [21] to high-level content descriptors such as acousticness, tempo or danceability (e.g., provided by the Spotify platform [1]). While these features are widely used to characterize single tracks, for a user model that captures the user's preferences well, these features have to be aggregated across all tracks the user has listened to. To this end, Pichl et al. [30] utilized content descriptors of tracks for representing a user's musical preference by computing the average acoustic features across all tracks the user has listened to. They also find that users create different playlists that feature different acoustic characteristics—implying that these playlists correspond to different sets of preferences of a user (which may naturally be context-related) and stress the need for more comprehensive user models to describe users' musical preferences [30]. Similarly, Wang et al. [36] state that people prefer different music for different daily activities. Along these lines, we argue that users may exhibit different preferences depending on the context and e.g., listen to more energetic music when doing sports or calming music when being at home [36]. These different preferences cannot be sufficiently reflected in a model that averages the characteristics of all the tracks a user listened to. In a probabilistic user model, Bogdanov et al. [4] characterize a user in a semantic feature space derived from low-level content features by utilizing Gaussian Mixture Models.

In this paper, we build upon and extend these previous works by proposing different user models to describe the musical preferences of users based on content descriptors of tracks. We perform a large-scale evaluation of these models in a track recommendation task based on 8 million listening events of 13,000 users. Our experiments show

Attribution: Eva Zangerle, Martin Pichl. "Content-based User Models: Modeling the many faces of musical preference", 19th International Society for Music Information Retrieval Conference, Paris, France, 2018.

[1] https://developer.spotify.com/web-api/get-several-audio-features/

that utilizing a user model based on a user's specific preferences regarding different types of music (modeled probabilistically by GMMs) complemented with a user's general musical preference achieves the best results. Our results show that in terms of recommendation quality, the proposed models contribute to substantially improved recommendation performance. We believe that our findings can contribute to improved user models for music recommender systems and generally, MIR tasks.

The remainder of this paper is organized as follows. Section 2 discusses related work and Section 3 presents the features utilized and the dataset underlying our experiments. Section 4 presents the user models proposed. Section 5 details the experimental setup and Section 6 presents the results of our study, which are discussed in Section 7. Section 8 concludes the paper and discusses future work.

2. RELATED WORK

Generally, Schedl et al. [32, 33] note that the user and his/her preferences are often not considered when it comes to MIR and MRS tasks. Particularly, the authors lay out that user modeling for such tasks has hardly been explored and evaluated yet.

To this end, content descriptors have widely been used in MIR and MRS. For similarity search, often a content-based similarity measure is used for matching queries and a music database [9, 20, 35, 39]. In the context of music recommender systems, Yoshii et al. [38] propose a hybrid recommender system that combines collaborative filtering via user ratings and content-based features modeled via Gaussian Mixture Models over MFCCs by utilizing a Bayesian network. Also, Liu [20] investigates different distance metrics for content-based recommender systems. Recently, also deep learning-based hybrid MRS have also been proposed [37]. In regards to user modeling for MRS, Bogdanov et al. compute a user's musical preferences by a set of exemplary tracks that the user enjoyed. They model the user's preference in a latent semantic space based on a set of diverse content features and propose a set of similarity-based recommender systems. One system models a user by a Gaussian Mixture Model based on the proposed semantic audio feature space. The authors evaluated these recommender systems in a user experiment with twelve users. As for musical preferences of users, Pichl et al. found in a large-scale study of Spotify users that music streaming users listen to different types of music. Those types can be observed via k-means clustering of content descriptors of tracks. They also found that users organize their music in playlists based on these types and stress the importance of more comprehensive user models to describe users' musical preferences [30]. Along these lines, we specifically investigate user models that are solely based on content descriptors. We propose six user models and compare these in a large-scale offline study based on a recommendation task comprising 13,000 users and 8 mio. listening events.

3. DATASET AND FEATURES

The main data source used in our experiments is the publicly available LFM-1b dataset [31], which provides the full listening histories of 120,322 Last.fm users. For each listening event (i.e., a certain user listening to a certain track), information about the track, artist, album and user is available. Besides the information contained within the LFM-1b dataset, we also require content features to describe tracks. Following the lines of, e.g., [1, 25, 30], we propose to rely on the Spotify API [2] to gather the following content descriptors for each track:

1. *Danceability* describes how suitable a track is for dancing and is based "on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity."
2. *Energy* measures the perceived intensity and activity of a track. This feature is based on the dynamic range, perceived loudness, timbre, onset rate and general entropy of a track.
3. *Speechiness* detects presence of spoken words. High speechiness values indicate a high degree of spoken words (talk shows, audio book, etc.), whereas medium to high values indicate e.g., rap music.
4. *Acousticness* measures the probability that the given track is acoustic.
5. *Instrumentalness* measures the probability that a track is not vocal (i.e., instrumental).
6. *Tempo* quantifies the pace of a track in beats per minute.
7. *Valence* measures the "musical positiveness" conveyed by a track (i.e., cheerful and euphoric tracks reach high valence values).
8. *Liveness* captures the probability that the track was performed live (i.e., whether an audience is present in the recording).

These features are high-level descriptors of the acoustic content of tracks. We argue that they are nevertheless representative and hence, the obtained results should give a good impression on the differences of the user models. We expect our findings to also hold for more complex and lower-level content descriptors such as e.g., Mel-Frequency Cepstral Coefficients (MFCC) [21].

To obtain these features for all tracks of the dataset, we apply the following steps: we perform a conjunctive query for the <track, artist, album>-triples extracted from the LFM-1b dataset using the Spotify search API [3] to gather the Spotify URI of each track. This URI is subsequently used to query the acoustic features API [4]. Finally, we add tracks for which can obtain all required features to the dataset [5]

Since the set of tracks a user listened to may also contain outlier tracks that may distort the user profile, we propose to remove outlier tracks from this set by apply-

[2] A detailed description of these features and the API can be found at https://developer.spotify.com/web-api/get-several-audio-features/.

[3] https://developer.spotify.com/web-api/search-item/

[4] https://developer.spotify.com/web-api/get-several-audio-features/

[5] Except for tempo, all of these features are given in the range of [0, 1] and for tempo, we apply a linear min-max scaling.

Item	Value
Listening Events (LE)	8,457,205
Users	12,995
Tracks distinct	965,293
Min. LE per User	1
Q_1 LE per User	252
Median LE per User	478
Q_3 LE per User	826
Max. LE per User	21,660
Avg. LE per User	650.80 (± 713.99)

Table 1. Dataset statistics.

ing the median absolute deviation (MAD) outlier detection method [19]. We consider a feature value an outlier if it is not within $M \pm a \cdot MAD$, where M is the median of this particular feature across all tracks of a user and MAD is the median absolute deviation of these values. We consider a value an outlier if it is not within within three MADs around the median, setting a rather conservative threshold $a = 3$ as proposed by [19]. Lastly, a track is considered as an outlier in the list of tracks of a particular user if one of its features is considered an outlier and consequently removed from the user listening history.

Applying this procedure results in a dataset of 55,149 users, 394,944,868 listening events and 3,478,399 distinct tracks. We randomly sample users from this dataset for our experiments, where we require each user to have more than 100 listening events to ensure that our user models are representative. We present basic statistics about the resulting dataset in Table 1. As can be seen, on average, each user has listened to 651 tracks.

4. USER MODELS

In the following, we present the proposed user models to capture user's listening preferences. We specifically focus on modeling users solely by acoustic features of tracks they listened to and deliberately neglect other information that could contribute to a user model (e.g., demographic user aspects, cultural information or further contextual features that might improve MRS and MIR performance).

4.1 Feature Space

Based on the users, tracks and their acoustic features within the dataset, we perform the following steps prior to the computation of the user models. Most of the proposed models require clustering tracks based on their acoustic features to find groups of tracks that exhibit similar features. Given that we aim to perform a large-scale analysis of the proposed user models (we perform the analysis on 8 million tracks and 13,000 users), these clustering computations are computationally intensive. Hence, we firstly perform a proximity-preserving dimension reduction on the input data by applying UMAP (Uniform Manifold Approximation and Projection) [23]. Also, the use of latent representations of elements in the musical ecosystem (users, tracks, etc.) has been to be effective in MIR and MRS tasks [18, 26, 27]. In our experiments, we compute a 2-d latent representation of tracks for the computation of user models. This allows us to inspect the resulting clusters visually during the development of the user models and, more importantly, reduces cluster computation time substantially, which naturally permits better scalability for larger datasets.

4.2 User Models

For modeling user preferences for musical tracks and their characteristics, we naturally require models for both tracks and users as we utilize a user's model and compare it with track models to find suitable similar tracks that may be recommended to the user.

As for modeling tracks and their characteristics, we rely on their acoustic features (AF; e.g., danceability or tempo). However, for users we require more sophisticated user models, as these have to represent a possibly extensive and diverse set of tracks and their characteristics to eventually represent a user's musical preferences. We propose user models that are based on clusters of similar tracks and utilize a user's membership in these clusters (i.e., the fact that user has listened to tracks that belong to a given cluster) to get a fine-grained representation of the many faces of the listening preferences of a given user. For determining such clusters and computing the membership of tracks in these clusters, we experiment with two approaches: (i) we utilize k-means clustering to find tracks that exhibit similar acoustic features and use the characteristics of these clusters to characterize users; and (ii) we apply Gaussian Mixture Models (GMM) [24] as these allow to model a track by the computed probability density function regarding the GMM's components. Based on a track's density functions, we derive a set of GMM-based user models. Generally, the idea is that based on these clusters or components, we aim to model a user based on the characteristics of one or multiple of these track clusters.

In the following, we describe the proposed user models to capture the musical preferences of users. An overview of the user models and the features used to characterize users and tracks is shown in Table 2.

Content avg: In a baseline model, we utilize the eight acoustic features of all tracks a user has listened to and compute the average across all tracks of a user for each of the features presented in Section 3. This allows us to describe a user with his/her average listening behavior, breaking a user's preferences down into eight acoustic features. Please note that in the remainder of this paper, we refer to models as *Content*-models if the representation of the user or a track relies on acoustic features.

Content avg, sd: This model is built upon the Content avg model, which we extend by adding the standard deviation of each of the acoustic features across all tracks of a user. We expect the added SD to mitigate the effects of averaging a large number of features that potentially differ substantially as users may listen to music with highly diverse acoustic characteristics. We again consider this model a baseline that additionally quantifies to which

Model	User Features	Track Feat.
Content avg	user AF avg	AF
Content avg, sd	user AF avg and SD	AF
Content binary k-means	avg. AF of single cluster	AF
Content weighted k-means	weighted avg. AF of clusters	AF
GMM	avg. densities of user's tracks	GMM densities
Content binary GMM	avg. AF of single GMM comp.	AF
Content weighted GMM	weighted avg. AF of GMM comp.	AF
GMM + Content avg, sd	GMM and user AF avg and SD	GMM, AF

Table 2. Overview of evaluated models (AF stands for acoustic features, GMM for Gaussian Mixture Model and SD for the standard deviation).

extent the user's musical preferences vary regarding the acoustic features of his/her listening history.

Content binary k-means: In this model, we rely on the clusters computed by a k-means clustering of all tracks within the dataset in the computed 2-d latent space. In a next step, we attribute each of the tracks a user has listened to a cluster and do a majority vote on the clusters to obtain the cluster that holds most of the user's tracks. We subsequently model a user using the characteristics of the cluster that contains the majority of the user's track. To represent this cluster, we compute the average of the eight acoustic features of all tracks contained in the cluster and add the according standard deviations. Single tracks are represented by its acoustic features. We consider this a rather simple model as we assign the user to a single cluster and hence, limit the model to a single preference scope.

Content weighted k-means: The previous model is limited as it is restricted to a single preference scope. To tackle this problem, we propose the Content weighted k-means model in which we now aim to address multiple sets of preferences of a user. Therefore, we again rely on the k-means clusters, however, we compute a weight for each cluster based on the number of tracks a user has listened to in each cluster. Based on the user's weights for each cluster, we compute a weighted average for each acoustic feature to represent the user, where each cluster is again characterized by its average acoustic features and its standard deviation. Again, in this model each track is represented by its acoustic features.

GMM: In this model, we utilize a Gaussian Mixture Model [24] for representing both the track and the user. Therefore, we compute Gaussian components and represent a track by its probability densities regarding the GMM components. For users, we compute the average probabilities for each component across all of the user's tracks to model a user's musical preferences by using the GMM components. We consider this model a proxy, as it does not directly utilize acoustic features to represent a track, but the probabilistic assignments of a track to a set of groups of tracks (components).

Content binary GMM: In contrast to the pure GMM model, this model relies on content features instead of probability densities to represent a user. Analogously to the Content binary k-means model, we rely on GMM to assign the user's tracks to components. In particular, we assign the tracks found in the user's listening history to GMM components. In a next step, we select the component with the highest number of user tracks assigned to, where we assign a track to the component with the highest probability density for the track. The user is then modeled by the characteristics of the selected component (again using the average and standard deviation across all acoustic features of the tracks assigned to the component), whereas each track is again represented by its acoustic features.

Content weighted GMM: This model is again analogous to the content weighted k-means model. However, we rely on a GMM to assign a user's tracks to certain a component as described in the previous model. Based on these assignments, we analogously compute the weighted mean and standard deviation for each acoustic feature for each GMM cluster to represent a user and the characteristics of tracks are captured by their acoustic features.

GMM + content avg, sd: In this model, we combine the GMM components baseline model with the content avg, sd baseline model and hence, represent a user by his/her component weights regarding the Gaussian Mixture Model and further add the average and standard deviation across all acoustic features of the user's tracks. Similarly, a track is represented by its GMM densities and its acoustic features.

We also performed experiments on representing users and tracks with cluster or component assignments only and did an analysis of further combinations of the proposed models. However, the results were below the evaluated baselines and hence, we do not list these models here.

5. EXPERIMENTAL SETUP

We model the evaluation of the proposed user models as a recommendation task, where we aim to obtain a ranked list of tracks that are of interest to the user. For this task, we rely on Gradient Boosting Decision Trees. Particularly, we utilize the popular XGBoost system [8], a scalable end-to-end tree boosting approach that has been shown to be highly suited for recommendation tasks [2, 28]. For the training phase of the tree, we set the training objective to be the binary classification error rate (i.e., the number of wrongly classified tracks in relation to all tracks classified, where tracks with a predicted probability of relevance larger than 0.5 are classified as relevant for the given user, and all other tracks are considered irrelevant for the user). Please note that we deliberately chose a classification-based recommendation approach and refrained from utilizing more elaborate recommender approaches such as context-aware matrix factorization [3] or tensor-based factorization approaches [15] as we aim to focus on user modeling aspects in this paper.

For the recommendation task carried out, we require a rating for each track in the dataset to define whether a given track was listened to and thus, considered relevant for a given user. Hence, we add a binary factor $rating$ to the processed dataset: for each unique $<user, track>$-combination, the $rating$ $r_{i,j}$ is 1 if the user u_i has listened to track t_j. Due to a lack of publicly available data, our dataset does not contain any implicit feedback of users (i.e., skipping behavior, session durations or dwell times during browsing the catalog). This is why we cannot estimate any preference towards a track a user not listened to as proposed by [14]. Thus, we assume tracks the user has not listened to as negative examples [14] and hence, assign a rating of 0 to these tracks. Even though there is a certain bias towards negative values as some missing values might be positive, Pan et al. [29] found that this method for rating estimation works well. To perform the proposed recommendation task via classification, we require the dataset to also include negative examples. Therefore, for each user, we add random tracks the user did not interact with (i.e., tracks t_j with $r_{i,j} = 0$ for the given user u_i) to the dataset until both the training and test sets are filled with 50% relevant and 50% non-relevant items. We chose to oversample the positive class to avoid class imbalance and hence, a bias towards the negative class.

Using the resulting data set, we train a XGBoost model that performs a binary classification on the relevance of tracks for a given users. We extract the probabilities underlying the classification decision to rank tracks by their probability of relevance in the recommendation task.

To evaluate the performance of the proposed user models in regards to recommendation quality, we perform a per-user evaluation. Therefore, we use each user's listening history and perform a *leave-k-out* evaluation (also known as hold-out evaluation) [6, 10] per user. Based on the dataset that now contains both positive and negative samples for each user, we compute a hold-out set of size k: along the lines of previous research [12, 13], we randomly select 10 positive samples (tracks that the user has listened to) and 100 negative samples (tracks the user has not listened to). These 110 tracks form the test set for each user, whereas the recommender system is trained on the remainder of the dataset. We compute the predicted ratings for the tracks in the test set and rank the track recommendation candidates w.r.t. the probability that the current track belongs to the positive class in descending order. For our experiments, we consider all predicted probabilities > 0.5 as a predicted interaction and thus, we consider these items as relevant, all others as irrelevant and hence, not added to the list of recommendations. [6]

Based on the predicted ratings, we compute *precision*, *recall*, and the F_1-measure to assess the top-10 accuracy [11]. We evaluate the 10 top ranked tracks as too many track recommendations might provoke choice overload and hence, is not feasible. The problem of choice overload has been addressed by Bollen et al. [5] who state that user satisfaction is highest when presenting the user with Top-5 to Top-20 items—naturally assuming that the recommendation list contains a sufficient number of relevant items for the user. For assessing the overall *precision*, *recall*, and F_1-measure of the evaluated recommender systems, we compute the measures for each individual user and compute the average among all users. For computing the *recall* measure, all relevant items in the test set are considered, independent of the number of recommendations. Thus, there is a natural cap for *recall*, namely the number of recommendations divided by the number of relevant items in the test set.

For the tuning of XGBoost parameters, we did a preliminary cross-evaluation aiming to optimize precision values for the proposed models and hence, set the number of maximum trees to learn the models to 2,000. For all other parameters, we relied on the default settings. For the training and tuning of k-means and GMM for the creation of the user models, we performed the following steps. For k-means, estimated the number of clusters by utilizing the elbow method based on the within-cluster sum of squares. For the given dataset, we estimated the number of clusters to be 5. For the GMM, we performed a training phase based on expectation maximization and determined the number of components using the Bayesian Information Criterion (BIC), which resulted in a total of 9 components for the GMM.

6. RESULTS

We present the results of our evaluation for a recommendation list of size ten in Table 3 and in a precision-recall plot depicted in Figure 1.

The best results are obtained by the GMM + Content avg, sd model, reaching a precision@10 of 0.771 and a recall@10 of 0.427 and hence, achieving substantially higher precision and recall scores than any other model. Comparing the results of this model to the GMM model (relying on solely the assignments to GMM components) and the Content avg, sd baseline model shows that those two models individually perform substantially worse than when combined. When inspecting the results of the GMM model, we find that solely relying on the GMM density functions does not suffice to represent a user's musical taste. Particularly, all content-based GMM or k-means models achieve higher performance when applied in isolation. However, combining a simple content-based approach that provides acoustic features regarding the user's general preferences, with GMM, provides us with a representative user model. This suggests that the GMM model captures a user's diverse preferences regarding the detected components and hence, his/her distribution in preference towards specific types of music, while his/her general preferences are captured by the average acoustic features and the according standard deviation.

Model	**Prec**	**Rec**	**F_1**
GMM + Content avg, sd	**0.771**	**0.427**	**0.632**
Content k-means weighted	0.606	0.316	0.400
Content k-means binary	0.573	0.300	0.383
Content binary GMM	0.569	0.298	0.381
Content weighted GMM	0.569	0.298	0.381
GMM	0.231	0.122	0.226
Content avg, sd	0.161	0.089	0.241
Content avg	0.159	0.087	0.241

Table 3. Precision, Recall and F_1@10, ordered by F_1.

[6] This distinction between the two classes is also utilized by XGBoost for binary classification tasks based on logistic regression.

Our results also show that the user models based on k-means clusters slightly outperform the methods based on GMM components (1.8% in recall, 3.7% in precision). Please note that for k-means we determined the number of clusters to be five, whereas we created nine GMM components (as described in Section 5). Our findings regarding the number of clusters are also in line with previous analyses on playlists [30], where the authors found that clustering the tracks within playlists into five clusters allows for cohesive and homogeneous clusters.

The weighted k-means approach achieves better results than the binary k-means approach. This seems natural as the former incorporates the user's membership in all clusters, whereas the latter does a majority vote and utilizes the resulting (single) cluster to characterize the user. However, this does not hold for the GMM-based approaches. While the differences between the weighted and binary k-means approaches are marginal, for GMM there is no difference between weighted and binary Content GMM.

The proposed baseline model Content avg achieves the lowest values regarding recall, precision and F_1. Adding the standard deviation to this model hardly impacts the results. We initially suspected that adding the SD to the model may allow mitigating the effects of aggregating possibly highly different tracks as we aggregate across all tracks of a user (regarding their acoustic features), however, this is not confirmed by our experiments. In preliminary experiments, we also used different representations of clusters: while we now utilize the mean acoustic features and the according SDs, we also used only the mean features. We found that the SD contributes only marginally as the dispersion of tracks in regards to acoustic features is already captured by the individual clusters/components and hence, the tracks contained in a single cluster/component are more homogeneous. We also experimented with models that utilize user-cluster assignments for k-means, however, those models achieved inferior results. In contrast, representing those clusters by the average acoustic features across all contained tracks seems to be representative. Combining k-means cluster assignments with content-based models also lead to inferior results, which we lead back to the fact that the GMM probability densities provide more information than sheer cluster-assignments.

Generally, we conclude that content features strongly contribute to user models and that grouping tracks into clusters (k-means) or components (GMM) and solely relying on the assignment to those clusters or components is not sufficient for a representative user model. Finding groups of similar tracks to represent users by user-group assignments via the tracks a user listened to is not expressive enough. Naturally, utilizing content features allows to compute higher-dimensional similarities between users and their tracks (in our experiments, 8 dimensions) and hence, a more fine-grained notion of similarity.

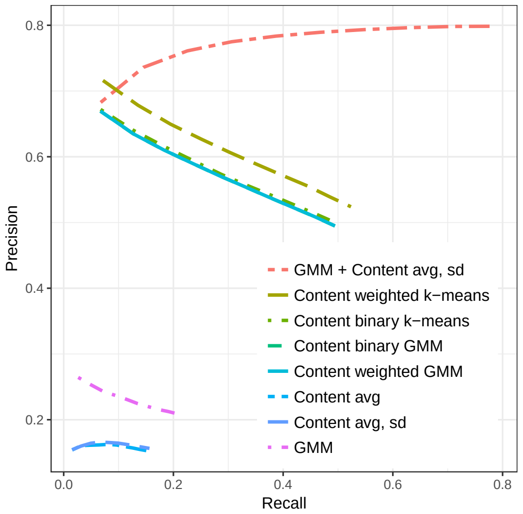

Figure 1. Precision-Recall curves for all models.

7. DISCUSSION

We find that a GMM that captures the specific preferences of a user towards a set of nine types of music (captured by nine GMM components) complemented by the general musical preference of a user (captured by the avg. acoustic features of his/her tracks) provides the best results.

Regarding the limitations of this study, we note that the content descriptors utilized are aggregated high-level features. This allowed us to keep the feature space smaller and to specifically focus on the user modeling aspects. Furthermore, this evaluation is solely based on aspects related to the content of tracks and no further user-related aspects as e.g., proposed by Schedl et al. [34]. Lastly, while the proposed models characterize users based on their interest in different clusters/components and hence, are able to build more specific user models, we still represent each cluster/component by the mean acoustic features of the tracks contained, which naturally limits the user model's specificity. However, we believe that our findings are a valuable contribution to advance user modeling for MIR and MRS and to foster further research in this direction.

8. CONCLUSION AND FUTURE WORK

We proposed and evaluated a set of user models for describing the musical preference of users by leveraging content descriptors of tracks the user has listened to. We find that a GMM complemented by the user's general musical preferences describes a user's different musical preferences best. We believe that our findings can contribute to improved user models for music recommender systems and generally, MIR tasks. In future work, we aim to investigate methods to combine the models evaluated by e.g., ensemble methods. Furthermore, we aim to tackle the problem that our current model still computes average acoustic features across a large number of tracks.

9. REFERENCES

[1] Jesper Steen Andersen. Using the echo nest's automatically extracted music features for a musicological purpose. In *2014 4th International Workshop on Cognitive Information Processing (CIP)*, pages 1–6, 2014.

[2] Takashi Ayaki, Hidekazu Yanagimoto, and Michifumi Yoshioka. Recommendation from access logs with ensemble learning. *Artificial Life and Robotics*, 22(2):163–167, 2017.

[3] Linas Baltrunas, Bernd Ludwig, and Francesco Ricci. Matrix factorization techniques for context aware recommendation. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, pages 301–304. ACM, 2011.

[4] Dmitry Bogdanov, Martín Haro, Ferdinand Fuhrmann, Emilia Gómez, and Perfecto Herrera. Content-based music recommendation based on user preference examples. In *4th ACM Conference on Recommender Systems. Workshop on Music Recommendation and Discovery (Womrad 2010)*, page 33, 2010.

[5] Dirk Bollen, Bart P. Knijnenburg, Martijn C. Willemsen, and Mark Graus. Understanding choice overload in recommender systems. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, pages 63–70, New York, NY, USA, 2010. ACM.

[6] John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 43–52. Morgan Kaufmann Publishers Inc., 1998.

[7] Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.

[8] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016.

[9] Parag Chordia, Mark Godfrey, and Alex Rae. Extending content-based recommendation: The case of indian classical music. In *Eight International Society for Music Information Retrieval Conference*, pages 571–576, 2008.

[10] P. Cremonesi, R. Turrin, E. Lentini, and M. Matteucci. An evaluation methodology for collaborative recommender systems. In *2008 International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution*, pages 224–231, 2008.

[11] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, pages 39–46, New York, NY, USA, 2010. ACM.

[12] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web*, pages 278–288, 2015.

[13] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182, 2017.

[14] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272. IEEE, 2008.

[15] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, pages 79–86. ACM, 2010.

[16] Yehuda Koren and Robert Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*, pages 145–186. Springer US, Boston, MA, 2011.

[17] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[18] Mark Levy and Mark Sandler. Learning latent semantic models for music from social tags. *Journal of New Music Research*, 37(2):137–150, 2008.

[19] Christophe Leys, Christophe Ley, Olivier Klein, Philippe Bernard, and Laurent Licata. Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. *Journal of Experimental Social Psychology*, 49(4):764 – 766, 2013.

[20] Ning-Han Liu. Comparison of content-based music recommendation using different distance estimation methods. *Applied Intelligence*, 38(2):160–174, Mar 2013.

[21] Beth Logan et al. Mel frequency cepstral coefficients for music modeling. In *Proceedings of the 1st International Symposium on Music Information Retrieval*, volume 270, pages 1–11, 2000.

[22] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In *Recommender Systems Handbook*, pages 73–105. Springer US, Boston, MA, 2011.

[23] L. McInnes and J. Healy. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *ArXiv e-prints*, 2018.

[24] Geoffrey McLachlan and David Peel. *Finite mixture models*. Wiley Series in Probability and Statistics, 2004.

[25] Matt McVicar, Tim Freeman, and Tijl De Bie. Mining the correlation between lyrical and audio features and the emergence of mood. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 783–788, 2011.

[26] Joshua L Moore, Shuo Chen, Thorsten Joachims, and Douglas Turnbull. Learning to embed songs and tags for playlist prediction. In *Twelth International Society on Music Information Retrieval Conference*, pages 349–354, 2012.

[27] Joshua L Moore, Thorsten Joachims, and Douglas Turnbull. Taste space versus the world: an embedding analysis of listening habits and geography. In *Fourteenth International Society for Music Information Retrieval Conference*, pages 439–444, 2014.

[28] Andrzej Pacuk, Piotr Sankowski, Karol Wegrzycki, Adam Witkowski, and Piotr Wygocki. Job recommendations based on preselection of offers and gradient boosting. In *Proceedings of the Recommender Systems Challenge*, pages 10:1–10:4. ACM, 2016.

[29] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 502–511, 2008.

[30] Martin Pichl, Eva Zangerle, and Gnther Specht. Understanding playlist creation on music streaming platforms. In *IEEE International Symposium on Multimedia*, pages 475–480. IEEE Computer Society, 2016.

[31] Markus Schedl. The LFM-1B Dataset for Music Retrieval and Recommendation. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 103–110, New York, NY, USA, 2016. ACM.

[32] Markus Schedl and Arthur Flexer. Putting the user in the center of music information retrieval. In *Twelth International Society for Music Information Retrieval Conference*, pages 385–390. Citeseer, 2012.

[33] Markus Schedl, Arthur Flexer, and Julián Urbano. The neglected user in music information retrieval research. *Journal of Intelligent Information Systems*, 41(3):523–539, 2013.

[34] Markus Schedl, Peter Knees, and Fabien Gouyon. New paths in music recommender systems research. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 392–393. ACM, 2017.

[35] B. Shao, D. Wang, T. Li, and M. Ogihara. Music recommendation based on acoustic features and user access patterns. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(8):1602–1611, Nov 2009.

[36] Xinxi Wang, David Rosenblum, and Ye Wang. Context-aware mobile music recommendation for daily activities. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 99–108, New York, NY, USA, 2012. ACM.

[37] Xinxi Wang and Ye Wang. Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pages 627–636, New York, NY, USA, 2014. ACM.

[38] Kazuyoshi Yoshii, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G Okuno. Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. In *Seventh International Society for Music Information Retrieval Conference*, 2006.

[39] Bingjun Zhang, Jialie Shen, Qiaoliang Xiang, and Ye Wang. Compositemap: A novel framework for music similarity measure. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 403–410, New York, NY, USA, 2009. ACM.