HYBRID MUSIC RECOMMENDER USING CONTENT-BASED AND SOCIAL INFORMATION

Paulo Chiliguano, Gyorgy Fazekas

Queen Mary, University of London
School of Electronic Engineering and Computer Science
Mile End Road, London E1 4NS, UK

ABSTRACT

Internet resources available today, including songs, albums, playlists or podcasts, that a user cannot discover require a tool for filtering the items that a user might consider relevant. Several recommendation techniques have been developed since the Internet explosion to achieve this filtering task. In an attempt to recommend relevant songs to users, we propose an hybrid music recommender system that considers real-world users information and high-level representation for audio data. The system uses a convolutional deep neural network to represent audio segments by genre and uses estimation of distribution algorithms to represent the user listening behavior. Finally, content-based filtering is employed to generate Top-N recommendations. Our proposed hybrid music recommender presents a convincing performance improvement over the content-based baseline model.

Index Terms— Deep Learning, Convolutional Neural Networks, Estimation of Distribution Algorithms, Recommender Systems, MIR.

1. INTRODUCTION

Recommender systems [1] are software or technical facilities to provide items suggestions or predict customer preferences by using prior user information. These systems play an important role in commercial applications to increase sales and improve user satisfaction. In general, recommender systems can be categorized in two major groups: collaborative filtering (CF) and content-based (CB) filtering. In CF [2, 3], a $m \times n$ rating matrix represents the relationships between m users and n items and recommendations are based on the computed similarities between rows (for users) or columns (for items). CF can be further subdivided in neighborhood models [4]. The recommendation process in CF is independent of the item features. However, CF ignores the items in the *long tail* [5] and it is unable to handle the *cold-start* [6] problem. On the other hand, CB filtering [7] is based on the analysis of the features that describe the items. The recommendation component consists in matching up the attributes of the items that a user has already rated, usually referred as the *user profile*, against the attributes of a previously unseen products to produce a list of *top-N* recommendations. The recommendation process in CB is entirely based on the attributes of the items, thus, the recommendations produced for each user is independent from the other users. Also, a CB recommender allows to recommend items that do not have any rating, therefore, the effects of cold-start problem can be diminished. However, the quality of recommendations from CB techniques depends on the size of historical dataset. A hybrid recommender [3] system is usually developed through the combination of CF and CB recommendation techniques to boost the advantages of CF by considering the feedback (rating) of users and the advantages of CB by taking into count the item attributes. Methods of combining recommendation techniques to accomplish hybridization can be classified in weighted, switching, mixed, feature combination, cascade, feature augmentation and meta-level methods.

In Section 2, we revise related work in music recommender systems. In Section 3, we present our proposed hybrid recommendation approach and its stages, also we describe the music genre representation of audio and user profile modeling. In Section 4, we present the experiments and evaluation protocols to assess the performance of the hybrid recommender. In Section 5, we discuss and analyze the results from the conducted experiments to evaluate the proposed hybrid music recommender. We conclude in Section 6, we present the conclusions and some thoughts for further research.

2. BACKGROUND

A hybrid music recommender system proposed by [8] considered the rating scores collected from Amazon.co.jp and acoustic features derived from the signals of musical pieces corresponding to Japanese CD singles that were ranked in weekly top-20 from April 2000 to December 2005. Acoustic features for each piece are represented as a *bag-of-timbres*, i.e., a set of weights of polyphonic timbres, equivalent to a 13-dimensional MFCC representation. Bags of timbres are computed with a Gaussian Mixture Model, considering the same combination of Gaussians for every musical piece. A music recommender proposed by [9] use a latent factor model for CB recommendation and the implementation of a convolutional neural network (CNN) [10] to predict the latent factors from music audio. To obtain 50-dimension latent vectors, they used a weighted matrix factorization (WMF) algorithm on the Taste Profile Subset [11]. Also, they retrieved audio clips for over 99% of the songs in the dataset from 7digital.com. To train the CNN, the latent vectors obtained through the WMF are used as ground truth. The input of the CNN is a log-compressed mel-spectrogram with 128 components computed from windows of 1,024 samples and a hop size of 512 samples (sampling rate of 22,050 Hz) of a 3 seconds window sampled randomly for each audio clip. They used 10-fold cross validation and obtained an average area under the ROC curve (AUC) of 0.86703 for prediction based on the latent factor vectors, outperforming the bag-of-timbres approach.

Inspired by natural selection of species, estimation of distribution algorithms (EDAs) [12, 13, 14] are optimization techniques that estimate a probabilistic model from a sample of promising individuals. The sample is used to generate a new population that satisfy a criteria, i.e., minimization of a *fitness* function. In Figure 1, we show the general flowchart of an EDA. Regarding to the types of distributions that an EDA are able to capture can be categorized in four broad groups: discrete variables EDA, permutation EDA, real-valued vectors EDA and genetic programming EDA.

In this paper, we present a hybrid music recommender to mitigate the cold-start problem in recommendation strategies. First, we use CNNs to describe the time-frequency content of an audio clip with a n-dimensional vector, whose dimensions represent the probability of a clip to belong to an specific music genre. Second, as a primary contribution, EDAs are investigated to model user profiles in terms of probabilities of music genres preferences. The algorithms use play count and the content vector of each song in the user's collection to optimize the profile. To our knowledge, this is the first approach that uses a continuous EDA for user profile modeling in recommender systems.

3. METHODOLOGY

The methodology used to develop our hybrid music recommender, shown in Figure 2 consists of four main stages. First, the collection of real world user-item data corresponding to the play counts of specific songs and the fetching of audio clips of the unique identified songs in the dataset. Secondly, the implementation of the CNN to represent the audio clips in terms of music genre probabilities as n-dimensional vectors. Next, permutation EDA and a continuous EDA are investigated to model user profiles based on the rated songs above a threshold. Finally, the process of top-N recommendation for the baseline and the hybrid recommender is described.

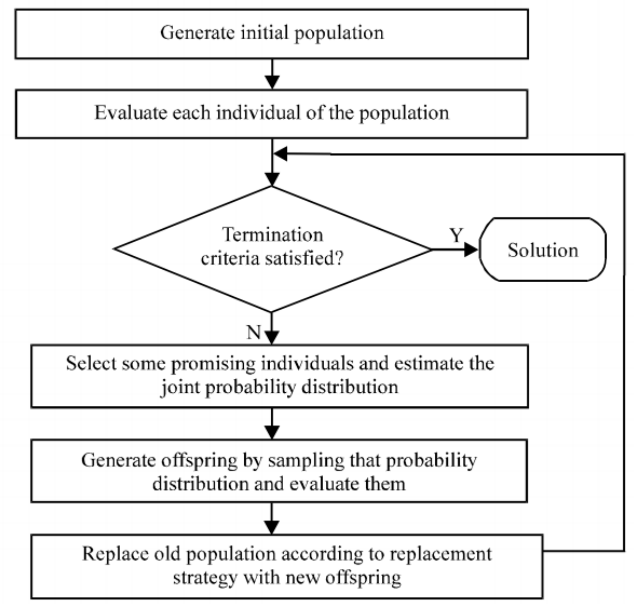
Generate initial population
Evaluate each individual of the population
Termination criteria satisfied?
Y
Solution
N
Select some promising individuals and estimate the joint probability distribution
Generate offspring by sampling that probability distribution and evaluate them
Replace old population according to replacement strategy with new offspring

Fig. 1. Flowchart of an EDA presented in [13]

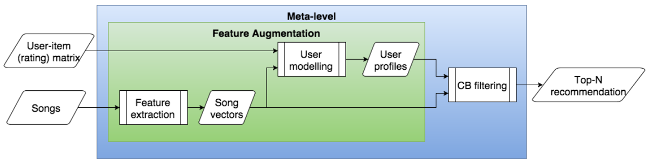

Fig. 2. Diagram of hybrid music recommender

3.1. Data collection

The Million Song Dataset [11] is a collection of audio features and metadata for a million contemporary popular music tracks which provides ground truth for evaluation research in Music Information Retrieval (MIR). This collection is also complemented by the Taste Profile subset which provides 48,373,586 triplets, each of them consist of anonymous user ID, Echo Nest song ID and play count. We choose this dataset because it is publicly available and it contains enough data for user modeling and recommender evaluation. However, it has potential mismatches between song ID and metadata, thus, we filter the triplets without errors in song IDs, obtaining 45,795,100 triplets. In addition, we reduce significantly the size of the dataset for experimental purposes and we consider only the users with more than 1,000 played songs and select the song identifiers of 1,500 most played songs. At the end, we obtain 65,327 triplets. For each song ID of the 1,500 most played songs, we retrieve the Echo Nest track ID and preview URL to fetch audio data from 7digital, resulting in a total of 640 available MP3 clips with a duration of 30 to 60 seconds (44.1 kHz, 128 kbps, stereo).

The reduced taste profile dataset represent the user listening habits as implicit feedback, i.e., play counts of songs, it is necessary to normalize the listening habits as explicit feedback, i.e., range of values $[1\ldots5]$ that indicate how much a user likes a song. Normalization of play counts is computed with the complementary cumulative distribution of play counts of a user, following the procedure given by [15].

3.2. Probability of music genre representation

Following the steps of [9], a segment equivalent to 3 seconds of each audio clip is loaded at a sampling rate of 22,050 Hz and converted to mono channel. For every segment, a mel-scaled power spectrogram with 128 bands is computed from windows of 1,024 samples with a hop size of 512 samples, resulting in a spectrogram of 130 frames with 128 components. Next, the spectrogram is converted to logarithmic scale in dB using peak power as reference. The spectrograms are normalized to have zero mean and unit variance in each frequency band. We use the GTZAN dataset [16] as ground truth for training, validation and testing the CNN.

In our project, we recreate a similar CNN architecture for music genre classification in [17]. A batch size of 20 and a dropout rate of 0.20 for the convolutional layer units are considered. The reshape of the spectrograms (130 frames$\times$128 frequency bands) to a 4-dimension tensor, compatible with the input of the first convolutional layer (batch size$\times 1 \times 130 \times 128$), is required. The first convolutional layer consists of 32 filters, each one with a size of 8 frames, with a max-pooling downsampling of 4, to reduce the size of the spectrogram along the time axis. The size of the resulting spectrogram is 30×128 and the output of this first convolutional layer is a 4-dimension tensor with a size of $20\times32\times30\times128$. The second convolutional layer consists of 32 filters, each one with a size of 8 frames, with a max-pooling downsampling of 4, to reduce the size of the spectrogram obtained in the first layer. The size of the new spectrogram is 5×128 and the output of this second convolutional layer is a 4-dimension tensor with a size of $20\times32\times5\times128$. Following the convolution process, the reshape of the 4-dimensional tensor of the output of the second convolutional layer is required to feed the fully connected multi-layer perceptron (MLP). The MLP consists of 500 ReLUs. Finally, the classification of music genre is accomplished with a logistic regression layer with 10 softmax units. Each output value corresponds to a music genre and each value represents the probability of a song to belong to a specific music genre.

3.3. User profile modeling with EDAs

In our project, we model each user profile following the EDA approach of [18] by minimizing a fitness function given by Equation (1)

$$fitness(profile_u) = \sum_{i \in S_u} \log(r_{u,i} \times sim(profile_u, t_i)) \tag{1}$$

where $sim(profile_u, t_i)$ is the cosine similarity between the profile of user *u* and the vector of probabilities of the song *i*, which is in the subset of relevant songs S_u, and $r_{u,i}$ is the rating of song *i* given by user *u*. In this approach, the probability values are discretized in 50 evenly spaced values over the interval $[0.1, 0.9]$.

Moreover, we extended the EDA approach using the continuous univariate marginal distribution algorithm $UMDA_c^G$ of [19].

3.4. Top-N songs recommendation

In our hybrid music model, the CB filtering module computes the similarity between a user interest profile and a each song vector in the test set. The songs are ranked in descending order and the first N songs of this list are recommended.

4. EXPERIMENTS

The hybrid music recommender system proposed in this project is evaluated through an off-line experiment and the results are presented with decision based metrics: Precision, recall, F1 measure and accuracy [15].

The reduced taste profile dataset is split in a training and a test set. For each user in the dataset, a random sample corresponding to 20 % of the total number of ratings is assigned to the test set, and the rest 80 % is assigned to the training set. The split process is iterated for 10 times, resulting in a total of 10 training and 10 test sets.

4.1. Top-N evaluation

For each song in the user test set, we look up if the song is included or not in the top-N recommendations. If the song rating is above the threshold (we find a rating ≥ 3 out of 5 suitable) and the song is ranked in the top-N recommendations, we count as a true positive, otherwise, if the song is not included in the recommendations we count as false negative. On the other hand, if the song rating is below the threshold and the song appears in the top-N recommendations, we count as a false positive, otherwise, if the song is not included on the recommendations, we count as a false negative.

5. RESULTS

In general, the results of the top-N evaluation demonstrate that the hybrid music recommender based on a permutation EDA presents a better performance compared with both the

CB recommender and the hybrid approach based on a continuous EDA. In Table 1, the results of top-5 recommendation are shown.

Recommender	Precision	Recall	F1	Accuracy
Content-based (baseline)	0.275 ± 0.087	0.010 ± 0.003	0.020 ± 0.007	0.681 ± 0.008
Hybrid (permutation EDA)	**0.391 ± 0.182**	**0.013 ± 0.007**	**0.025 ± 0.013**	**0.685 ± 0.009**
Hybrid (continuous UMDA)	0.318 ± 0.142	0.011 ± 0.005	0.021 ± 0.011	0.683 ± 0.009

Table 1. Evaluation of recommender systems (N=5)

In Table 2, the results of top-10 songs recommendation are shown. In this case, the precision value improves for the CB recommender. The accuracy values for all recommender systems tend to decrease.

Recommender	Precision	Recall	F1	Accuracy
Content-based (baseline)	0.301 ± 0.059	0.022 ± 0.007	0.041 ± 0.012	0.678 ± 0.007
Hybrid (permutation EDA)	**0.370 ± 0.073**	**0.024 ± 0.007**	**0.045 ± 0.013**	**0.682 ± 0.009**
Hybrid (continuous UMDA)	0.309 ± 0.100	0.019 ± 0.007	0.036 ± 0.013	0.679 ± 0.009

Table 2. Evaluation of recommender systems (N=10)

In Table 3, the results of top-20 songs recommendation are shown. In this case, the recall values rise for all the recommender systems, compared with the top-5 and top-10 recommendations, but the precision and accuracy tend to decrease. At this point, we can deduce that our hybrid recommender approaches could improve the recall without losing reached precision if N is in a value between 10 and 20.

Recommender	Precision	Recall	F1	Accuracy
Content-based (baseline)	0.281 ± 0.052	0.041 ± 0.006	0.071 ± 0.010	0.666 ± 0.006
Hybrid (permutation EDA)	**0.363 ± 0.041**	**0.047 ± 0.008**	**0.084 ± 0.014**	**0.676 ± 0.007**
Hybrid (continuous UMDA)	0.302 ± 0.067	0.039 ± 0.011	0.070 ± 0.019	0.671 ± 0.010

Table 3. Evaluation of recommender systems (N=20)

6. CONCLUSIONS AND FURTHER WORK

The whole aim of our project has been the design and the implementation of an hybrid music recommender in order to mitigate the cold-start problem in content-based recommender systems. We investigated several types of hybridization in recommender systems to choose a suitable architecture for the available datasets. To represent real world users and raw waveforms, we decided to investigate and implement state-of-the-art techniques. Despite of the success in computer vision field, we found in our project that convolutional deep neural networks achieve similar results to a long-established content-based music genre classifier in music information retrieval field. Due to the natural selection concept associated to estimation of distribution algorithms, we investigated and considered these optimization techniques for modeling users' listening behavior in terms of probabilities of music genres from the songs they have listened. On the other hand, we found that a limited number of genres for song representation lead us to coarse predictions according to decision-based metrics.

For the future, we have the intention to enhance our hybrid music recommender considering a wide range of music genres or latent vectors for item representation. We shall work on investigating several configurations of convolutional deep neural networks and different types of deep learning techniques, particularly, unsupervised learning approaches, for a better high-level representation of audio waveforms. In addition, we will continue investigating the fascinating estimation of distribution algorithms, considering another optimization functions, to model user profiles in recommender systems. Finally, we also consider the evaluation of the hybrid recommender with an online experiment.

7. REFERENCES

[1] Prem Melville and Vikas Sindhwani, "Recommender systems," in *Encyclopedia of machine learning*, pp. 829–838. Springer, 2010.

[2] L. Yao, Q.Z. Sheng, A.H.H. Ngu, J. Yu, and A. Segev, "Unified collaborative and content-based web service recommendation," *IEEE Transactions on Services Computing*, vol. 8, no. 3, pp. 453–466, 2015.

[3] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Modelling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[4] Y. Hu, C. Volinsky, and Y. Koren, "Collaborative filtering for implicit feedback datasets," *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 263–272, 2008.

[5] H. Yin, B. Cui, J. Li, J. Yao, and C. Chen, "Challenging the long tail recommendation," *Proceedings of the VLDB Endowment*, vol. 5, no. 9, pp. 896–907, 2012.

[6] C. Dai, F. Qian, W. Jiang, Z. Wang, and Z. Wu, "A personalized recommendation system for netease dating site," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1760–1765, 2014.

[7] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor, Eds., pp. 73–105. Springer US, 2011.

[8] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H.G. Okuno, "An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 2, pp. 435–447, 2008.

[9] Aaron van den Oord, Sander Dieleman, and Benjamin Schrauwen, "Deep content-based music recommendation," in *Advances in Neural Information Processing Systems 26*, C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, Eds., pp. 2643–2651. Curran Associates, Inc., 2013.

[10] Yoshua Bengio, Ian J. Goodfellow, and Aaron Courville, "Deep learning," Book in preparation for MIT Press, 2015.

[11] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere, "The million song dataset," in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[12] Martin Pelikan, Mark W Hauschild, and Fernando G Lobo, "Estimation of distribution algorithms," in *Springer Handbook of Computational Intelligence*, pp. 899–928. Springer, 2015.

[13] C. Ding, L. Ding, and W. Peng, "Comparison of effects of different learning methods on estimation of distribution algorithms," *Journal of Software Engineering*, vol. 9, no. 3, pp. 451–468, 2015.

[14] Roberto Santana, Concha Bielza, Pedro Larraaga, Jose A. Lozano, Carlos Echegoyen, Alexander Mendiburu, Rubn Armaanzas, and Siddartha Shakya, "Mateda-2.0: A matlab package for the implementation and analysis of estimation of distribution algorithms," *Journal of Statistical Software*, vol. 35, no. 7, pp. 1–30, 7 2010.

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

[16] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[17] Corey Kereliuk, Bob L. Sturm, and Jan Larsen, "Deep learning and music adversaries," *CoRR*, vol. abs/1507.04761, 2015.

[18] T. Liang, Y. Liang, J. Fan, and J. Zhao, "A hybrid recommendation model based on estimation of distribution algorithms," *Journal of Computational Information Systems*, vol. 10, no. 2, pp. 781–788, 2014.

[19] Marcus Gallagher, Ian Wood, Jonathan Keith, and George Sofronov, "Bayesian inference in estimation of distribution algorithms," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*. IEEE, 2007, pp. 127–133.