A Hybrid Approach for Music Recommendation

Lin Zhu
Ctrip Travel Network Technology Co., Limited.
Shanghai, P.R. China 200050
zhulb@Ctrip.com

Yihong Chen
Ctrip Travel Network Technology Co., Limited.
Shanghai, P.R. China 200050
yihongchen@Ctrip.com

Wen Jiang
Ctrip Travel Network Technology Co., Limited.
Shanghai, P.R. China 200050
w.jiang@Ctrip.com

ABSTRACT

The task of WSDM 2018 Music Recommendation Challenge is predict the probability of a user re-listening to a song within a specified time window after the first observable listening event. This paper presents our approach to this challenge. We built our recommendation models using multiple additive decision trees and factorization machines. By capturing the time-series characteristics of the music listening data, we can achieve significant improvement over baseline models. Meanwhile, ensemble of a collection of models that take into consideration the cold-start nature of the music recommendation task can further significantly improve upon the best single model. We show how our approach achieved an AUROC score of 0.73666 on the withheld test set, and thereby attaining the overall 5th place in the competition.

CCS CONCEPTS

•Information systems →Recommender systems;

KEYWORDS

Music recommender systems, challenges, WSDM cup

ACM Reference format:
Lin Zhu, Yihong Chen, and Wen Jiang. 2016. A Hybrid Approach for Music Recommendation. In *Proceedings of ACM Conference, Los Angeles, USA, Feb 2018 (WSDM Cup '18),* 4 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Thanks to the growing popularity of music streaming services such as Spotify, KKBOX, and Apple Music, music fans are now given access to tens of millions songs, from which they can potentially discover a wealth of new favorite songs. Given such overwhelming quantities of available listening choices, personalized machine learning algorithms have gradually assumed the roles that originally belong to Radio DJs, namely to provide listen suggestions to users and increase their engagement. So far, a large number of work have been done to develop music recommendation systems both in academia and industry [2, 3, 7, 21], nevertheless existing systems are still far from satisfactory and have much room for improvement[18]. Major unsolved issues include ones that are shared by recommendation systems in general, such as the so-called cold-start problem, i.e., due to lack of historical data, it is difficult or unreliable to apply such systems for new users or songs that are only recently added into the catalog[22], and ones that are specific to the music recommendation domain, such as the sequential nature of music assumptions and the appropriateness of repeated recommendations[18].

In this year, the WSDM Cup, a competition held annually as part of the prestigious ACM International Conference on Web Search and Data Mining (WSDM), challenges competitors from all over the world to help tackle the abovementioned problems and build a better music recommendation system. The used dataset is kindly donated by KKBOX, which is Asia's leading music streaming service and holds the worlds most comprehensive Asia-Pop music library with over 30 million tracks. In this challenge, the competitors are required to predict the probability of a user re-listening to a song within a specified time window after the first observable listening event.

To tackle this challenge, we have designed a hybrid approach to better handle the various unique characteristics of the WSDM KKBOX music data set. By capturing the time-series characteristics of the music listening data, we can achieve significant improvement over baseline models, meanwhile, ensemble of a collection of models that further take into consideration the cold-start nature of the music recommendation task can further significantly improve upon the best single model.

The paper is organized as follows. The KKBOX dataset is introduced in Section 2, Section 3 discusses the components of our approach, Section 4 reports the model performance, and the paper concludes in Section 5.

2 THE DATASET DESCRIPTION

There are several datasets provided by KKBOX for the competition:

- *APP information of listening events*: circumstances for a number of unique user-song listen events on the KKBOX mobile apps within a specific time duration, such as the name of the tab where the event was triggered (denoted as "system tab") and the name of the layout a user sees ("source screen name"), the training and testing data are obtained by dividing these listening events into 2 groups based on time.
- *Information of the users*, which includes the associated member id (referred to as "msno"), city id, age, registration method, registration time, and membership expiration date, etc.
- *Information of the songs*, which includes the song id, song length, genre ids, artist name, composer, lyricist,

WSDM Cup '18, Los Angeles, USA
2016. 978-x-xxxx-xxxx-x/YY/MM. . . $15.00
DOI: 10.1145/nnnnnnn.nnnnnnn

International Standard Recording Code (ISRC), and language information.

Given the datasets, the competitors were asked to predict the likelihood of a user listening to a song again within a month after the user's very first observable listening event of that song, and the evaluation metric is Area under Receiver Operating Characteristic curve (AUROC).

3 OUR APPROACH

3.1 Feature Engineering for User-Song Listening Events

A straightforward solution for this challenge is to tackle it as a classification problem: based on the information given for a user-song listening event, classify whether the user will re-listen the song. Based on this viewpoint, we spent considerable effort on extracting features of users, items and circumstances for every user-song listening event, hoping that integration of such comprehensive features lead to better performance of our method. These features can be categorized into the following 3 types:

3.1.1 Categorical Features. As is shown in Section 2, users, songs, and listening events are all supplied with side attributes in this competition, and since most of these side attributes are discrete-valued, examples of which include song id, genre id, artist, composer, lyricist, ISRC, and language information for each song, and member id, city id, age, registration method for each user, categorical features can be natural representations for them. Meanwhile, in order to better capture the interactions between users, KKBOX apps, and songs, we constructed additional combinations of these categorical features as new features: for example, the feature "msno+artist" denotes unique pair of interactions between a member and an artist, namely listening events where the member listens to songs of the artist. Additional features, such as "msno+composer" and "song id+system tab", are defined in a similar vein.

3.1.2 Simple Count-based Features. These type of features measure the frequencies of the categorical features in the training data, for example, "artist_count" measures how many times does the artist associated with a given listening event occurs in the dataset. Other count-based features, such as "msno+artist_count", "msno+composer_count", "system tab+composer_count", are similarly defined.

3.1.3 Time-series-based Features. As mentioned in Section 2, the given data are ordered by occurrence time, thus even though no explicit time stamps are given, one can still exploit the time-series-nature of the input data for performance improvement. The features that we constructed using this special property include:

- **"Age" features**: Unlike the standard setting of recommendation systems where the item set and user set are assumed to be fixed, the KKBOX dataset spans 2 years (2016 and 2017), during which newly released songs were constantly being added to the music repository, and new users would also come in continuously. On the one hand, recommending recently released content is important since users often prefer fresh content in practice[6], on the other hand, it is also important for bias correction to account for such "age" of songs and users, since it means that a large number of songs and users are not "evenly" distributed in the dataset. Based on these observations, we firstly construct the features "msno_occurence_so_far", "song_occurence_so_far", "artist_occurence_so_far", "composer_occurence_so_far", and "lyricist_occurence_so_far", which measure the occurrence time of the corresponding user/song/artist/composer/lyricist before the user-song listening event; similarly we also construct features such as "msno_occurence_remain", "song_occurence_remain", "artist_occurence_remain", "composer_occurence_remain", etc., which describe the occurence frequency of user/song/artist/composer in the dataset after the current listening event.
- **"Session" features**: for recommendation tasks in online streaming systems such as KKBOX, it is often beneficial to consider the interactions between a user and the system/items from a session-based viewpoint[16, 20]. A session is a group of interactions that take place within a given time interval[10]. For example, a user may sometimes listen to songs that share the same artist or style during a short time period, and identification of such sessions may improve the recommendation accuracy. Recall that in this competition, the time stamps of listening events are not given explicitly, we instead estimated sessions by greedily grouping adjacent listening records that belong to the same use together. Given the estimated sessions, we constructed 3 features: (i) "session_msno_count", which measures the number of sessions that a user played in the dataset; (ii) "avg_song_num_per_session_msno', which measures the average number of songs that a user played in a session; (iii) "current_session_count", which measures the number of songs that the current session contains.

3.1.4 Model Learning. With all features constructed, a wide variety of nonlinear/linear functions can be explored to fit the learning target. In this competition, we specifically consider gradient boosting decision trees (GDBT)[8, 9], a powerful machine-learning technique that has a wide range of successful applications[12, 26]. GDBT is also particularly suitable for dealing with the problem considered in this competition, as it can handle missing values elegantly and scale beyond billions of samples thanks to recent proposed algorithmic development[4, 11, 14]. We adopted the software package LightGBM[11] for GDBT model fitting, as it can directly handle the numerous categorical features constructed by us, without the requirement of cumbersome preprocessing steps such as one-hot encoding[15].

3.2 Embedding Method

In previous section, we tackle the predictive task of the challenge from the "classification" perspective, and directly use categorical features to represent different users/artists/composers, etc. However, it is known that models learned as such tend to only memorize interactions between categorical features that already exist in the training data[5]. On the other hand, embedding-based models, such as factorization machines[17] or deep neural networks[6] embed categorical IDs into a low-dimensional space to represent latent preferences, and can thereby generalize to previously un-observed categorical feature pairs[5]. Therefore, other than the feature-engineering-based classification models described above, we additionally adopted factorization machines to learn latent factors for each user and songs pairs to rank given songs based on the likelihood of observed ones. The provided metadata, such as age and city for a user, or language of a song, were also integrated into the model learning process. To fulfill our needs, We used the RankingFactorizationRecommender class in the Graphlab package[13]. We set the number of factors to 60 (num_factors), linear regularization value to be 10^{-4} (linear_regularization), interactions regularization value to be 10^{-5} (regularization), and the number of maximum iterations was set to be 10 (max_iterations).

3.3 Model Variants

In this section, we introduce several variants of the base models, which were combined with the based models to get the final predictive results.

3.3.1 Cold Start Simulation. As mentioned in the introduction, cold start poses serious challenges for recommendation systems in practice, and the music recommendation systems are not immune to this problem. For example, about 20% of the listening records in the testing set contain users or songs that don't occur in the training data, which may hurt the predictive performance.

Recall from Section 3.1 that in our model, users and songs are all represented as categorical features, thus the cold start problem in our context may be alternatively understood as the "curse of dimensionality" problem of categorical features with high cardinality[1]: due to limited size of the training set, it is not possible to observe every possible value of such categorical variables, therefore, if the learned model relies too much on information provided by these unreliable categorical features, it may not generalize well to future test data.

During the competition, we attempted to ameliorate this problem by borrowing ideas from denoising autoencoders[24] and dropout[19, 25], and retrained the GBDT model without high cardinality categorical features, which includes the user id ,song id, artist name, composer, and lyricist, etc. In this way, the model is forced to explain the observed data without the help of these unreliable features.

3.3.2 Drop-out Training of Multiple Additive Regression Trees. On the other hand, due to the iterative nature of GDBT's updating process, it has the problem that trees added at later iterations tend to only impact a small fraction of the training instances, which may have negative influence on the predictive accuracy[23]. An alternative for ameliorating this problem is to adopt the "dropout" training scheme proposed in [23], which randomly drops some of the learned trees when updating new trees. This functionality is also conveniently implemented in the LightGBM package, which was directly adopted in our experiments.

3.4 Model Ensemble

Based on the various models introduced above, our final prediction for a user-song listening event x is given by[1]:

$$\frac{1}{4}\begin{pmatrix} pred_{GDBT}(x) + pred_{GDBT+Cold}(x) + \cdots \\ \cdots pred_{DART}(x) + pred_{DART+Cold}(x) \end{pmatrix} + \frac{1}{5} pred_{FM}(x), \tag{1}$$

where the subscript "$_{Cold}$", and "$_{DART}$' ' denote the models trained using the methodologies introduced in Section 3.3.1 and 3.3.2, respectively, and "$_{FM}$" denotes factorization machine.

4 RESULTS

Table 1 lists the performance of various combinations of our previously described models. The used notations are the same as equation (1). As shown in the table, ensemble of these model can greatly improve the performance, and our overall approach achieved 5th place in the final leaderboard.

5 CONCLUSION

This paper presents our approach to the WSDM Challenge 2018. By capturing the time-series characteristics of the music listening data, we can achieve significant improvement over baseline models. Meanwhile, ensemble of a collection of models that further take into consideration the cold-start nature of the music recommendation task can further significantly improve upon the best single model. Our approach achieved 5th place in the competition.

REFERENCES

[1] Y Bengio, R Ducharme, P Vincent, and C Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research* 3, 6 (2003), 1137–1155.

[2] Chih-Ming Chen, Ming-Feng Tsai, Yu-Ching Lin, and Yi-Hsuan Yang. 2016. Query-based music recommendations via preference embedding. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 79–82.

[3] Shuo Chen, Josh L Moore, Douglas Turnbull, and Thorsten Joachims. 2012. Playlist prediction via metric embedding. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 714–722.

[4] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), 785 – 794.

[5] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, and others. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*. ACM, 7–10.

[1] Note that the online evaluation system of the competition requires that the predictions must take values between 0 and 1, thus a simple post-processing step is needed to meet this requirement

Table 1: Performance of Various Models

Model	AUROC	Leaderboard Position
$pred_{GDBT}(x)$	0.72558	8
$\frac{1}{2}\left(pred_{GDBT}(x) + pred_{DART}(x)\right)$	0.72781	7
$\frac{1}{4}\left(pred_{GDBT}(x) + pred_{GDBT+Cold}(x) + pred_{DART}(x) + pred_{DART+Cold}(x)\right)$	0.73505	5
$\frac{1}{4}\left(pred_{GDBT}(x) + pred_{GDBT+Cold}(x) + pred_{DART}(x) + pred_{DART+Cold}(x)\right) + \frac{1}{5}pred_{FM}(x)$	0.73666	5

[6] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 191–198.

[7] Sander Dieleman. 2016. Deep learning for audio-based music recommendation. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS'16). ACM*.

[8] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2001. *The elements of statistical learning*. Vol. 1. Springer series in statistics New York.

[9] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[10] Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 306–310.

[11] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*. 3149–3157.

[12] Jianxun Lian, Fuzheng Zhang, Min Hou, Hongwei Wang, Xing Xie, and Guangzhong Sun. 2017. Practical Lessons for Job Recommendations in the Cold-Start Scenario. In *Proceedings of the Recommender Systems Challenge 2017*. ACM, 4.

[13] Yucheng Low, Danny Bickson, Joseph Gonzalez, Carlos Guestrin, Aapo Kyrola, and Joseph M Hellerstein. 2012. Distributed GraphLab: a framework for machine learning and data mining in the cloud. *Proceedings of the VLDB Endowment* 5, 8 (2012), 716–727.

[14] Qi Meng, Guolin Ke, Taifeng Wang, Wei Chen, Qiwei Ye, Zhi-Ming Ma, and Tieyan Liu. 2016. A communication-efficient parallel algorithm for decision tree. In *Advances in Neural Information Processing Systems*. 1279–1287.

[15] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and others. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12, Oct (2011), 2825–2830.

[16] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks. *arXiv preprint arXiv:1706.04148* (2017).

[17] Steffen Rendle. 2010. Factorization machines. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 995–1000.

[18] Markus Schedl, Hamed Zamani, Ching-Wei Chen, Yashar Deldjoo, and Mehdi Elahi. 2017. Current Challenges and Visions in Music Recommender Systems Research. *arXiv preprint arXiv:1710.03208* (2017).

[19] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research* 15, 1 (2014), 1929–1958.

[20] Bartłomiej Twardowski. 2016. Modelling contextual information in session-aware recommender systems with neural networks. In *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, 273–276.

[21] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Advances in neural information processing systems*. 2643–2651.

[22] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. 2017. A Meta-Learning Perspective on Cold-Start Recommendations for Items. In *Advances in Neural Information Processing Systems*. 6907–6917.

[23] Rashmi Korlakai Vinayak and Ran Gilad-Bachrach. 2015. DART: Dropouts meet multiple additive regression trees. In *Artificial Intelligence and Statistics*. 489–497.

[24] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11, Dec (2010), 3371–3408.

[25] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. DropoutNet: Addressing Cold Start in Recommender Systems. In *Advances in Neural Information Processing Systems*. 4964–4973.

[26] Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. 2010. Adapting boosting for information retrieval measures. *Information Retrieval* 13, 3 (2010), 254–270.