Context-aware Music Recommender System Based on Implicit Feedback

Yasufumi Takama
Tokyo Metropolitan University
ytakama@tmu.ac.jp, https://krectmt3.sd.tmu.ac.jp

Jing-cheng Zhang
(affiliation as previous author)

Hiroki Shibata
(affiliation as previous author)
hshibata@tmu.ac.jp

keywords: music recommendation, context-aware recommendation, factorization machines, implicit feedback

Summary

This paper proposes a method for recommending music items without explicit feedback. Context and content features are used as auxiliary information to compensate for implicit feedback. The recent development of communication technology and portable electronic devices has changed the way of consuming music. We can access a vast amount of music items via online music streaming services. As a result, finding appropriate music items from enormous resources gets to be difficult for users. To help users discover their favorite music items, recommender systems in the music domain have been studied. This paper focuses on two challenges specific to music recommender systems: the difficulty of obtaining explicit feedback such as ratings, and the importance of making use of context information. To handle the context information as auxiliary information to compensate for implicit feedback, this paper employs FMs (Factorization Machines), in which the context information is treated as features. Utilizing the merit of FMs that can easily introduce features, this paper also introduces content features in addition to context features. As it is known that using low-level content features directly is not effective because of the semantic gap, this paper proposes two types of abstract content features: UGP (user genre profile) and UCP (user content profile). The effectiveness of the proposed method and the effect of negative sampling methods are evaluated in terms of MPR with #nowplaying-rs and LFM-1b dataset. The result of the experiment shows that the proposed method outperforms wALS (weighted Alternating Least Squares), which is one of the popular music recommendation algorithms based on matrix factorization. The characteristics of the proposed sampling methods are investigated with different settings of the parameters and the ratio of negative samples. As for the effectiveness of each feature, it is found that a feature is effective when JS (Jensen-Shannon) divergence of popularity distribution among different feature values is large. It is also shown that the UCP and UGP cluster labels are more effective than using content features directly.

1. Introduction

This paper proposes a music recommender system, which is based on implicit feedback and utilizes context/content information. The recent development of communication technology and portable electronic devices has changed the way of consuming music. We need not buy a CD from traditional brick-and-mortar shops to play it with CD players or export songs to digital devices. Instead, we can enjoy music easily regardless of time and place by accessing online music streaming services such as Spotify\(^1\), Apple Music\(^2\), and Amazon Music\(^3\). These up-to-date music streaming services offer thousands of tracks in a wide variety of music genres via Internet-accessible smartphones or tablets. However, finding appropriate music items from enormous resources gets to be difficult for users. To help users discover their favorite music items, recommender systems in the music domain have been studied [Adomavicius 15, Celma 10, Deng 15, Oord 13].

This paper focuses on two challenges music recommender systems should tackle. First, different from such domain as movie and e-commerce, explicit feedback, i.e. rating, is not usually available. Instead, the interaction between users and items is recorded as a LE (listening event), a record of context information when a user listened to a music item. Therefore, a music recommender system

---

*1 https://www.spotify.com/jp/
*2 https://www.apple.com/jp/apple-music/
*3 https://www.amazon.co.jp/gp/dmusic/promotions/PrimeMusic
should estimate a user’s preference without explicit feedback.

Second, it has been shown that context information such as time, weather, demographic information, and the emotional state of the user plays an important role in the music domain [Kaminskas 12]. How to utilize such context information is crucial for music recommendation.

To handle the context information as auxiliary information to compensate for implicit feedback, this paper employs FMs (Factorization Machines) [Rendle 12], in which the context information is treated as features. It has been shown that FMs can flexibly consider the interaction of users, items, and context features with lower time complexity than existing methods using a tensor factorization approach [Adomavicius 15].

Utilizing the merit of FMs that can easily introduce features, content features are also introduced in this paper in addition to context features. As will be noted in Section 2.2, using low-level content features directly is not effective because of the semantic gap [Celma 10]. Therefore, this paper proposes two types of content features, UGP (user genre profile) and UCP (user content profile), which are generated from the content features available in a dataset.

FMs need both positive and negative samples. A straightforward approach is to use LEs as positive samples, and all the combination of context and music items a user has yet to be experienced as negative samples. The problem of such an approach is that the number of negative samples is much larger than that of positive ones, which causes a problem of time complexity especially when training FMs. To reduce the number of negative samples, this paper employs three types of negative sampling methods: (1) Random Sampling (RS), (2) Top Discount Popularity Sampling (TDPS), and (3) Priority Popularity Sampling (PPS). The RS selects negative samples randomly from music items that are not played by a user under the same context as actual LEs (positive samples). The TDPS and PPS consider the popularity of music items when determining negative samples. The TDPS divides items into popular/unpopular groups, to each of which different probability is given. The PPS employs non-uniform probability distribution of selecting a music item as a negative sample based on its popularity.

The effectiveness of the proposed method is evaluated with an offline experiment using two datasets: #nowplaying-rs [Poddar 18] that is generated from the tweets with hashtag #nowplaying posted in 2014, and LFM-1b dataset [Schedl 16] generated by collecting LEs from Last.fm. The results show that the proposed method outperforms wALS (weighted Alternating Least Squares) [Hu 08], which is one of the popular music recommendation algorithms based on matrix factorization.

It is confirmed that the performance of recommendation could be improved by using appropriate context/content features. The experimental results show that Language and UCP are especially effective for #nowplaying-rs dataset, and the combination of Nationality and UGP is effective for LFM-1b dataset. The proposed system is designed by combining several techniques considering the above-mentioned challenges specific to music recommendation. The findings of the experiments on the two large music datasets are expected to be valuable to the development of music recommender systems.

This paper extends the research reported in the JSAI2019 post-proceedings [Zhang 20] by adding an experiment with LFM-1b dataset, the proposal of UGP and UCP, detailed analysis of the experimental results such as the effect of context/content features. The organization of the paper is as follows. Chapter 2 introduces existing works on music recommendation, which is followed by the description of the proposed method in Chapter 3. Experimental results are shown in Chapter 4.

2. Related Work

2.1 Music recommendation

In the case of music recommendation, explicit feedback such as rating is not usually available. Instead, the interaction between users and items is recorded as a LE. Therefore, how to use such implicit feedback from users effectively has been a crucial theme for music recommendation [Oard 98].

The most common approach of CF (collaborative filtering) is a matrix factorization-based approach [Rendle 09, Salakhutdinov 07]. Hu, et al. have proposed a matrix factorization-based CF called wALS for music recommendation [Hu 08]. They calculates a play count of a music item from implicit feedback data and used it for recommendation. As a play count is not as reliable as explicit rating, they introduced a confidence computation to a loss function based on the playing count as shown in Eq. (1) and considered all missing LEs as negative samples.

\[
L(X,Y) = \sum_{x_i \in X, y_j \in Y} (1 + \alpha r_{i,j})(p_{i,j} - x_i^T y_j)^2 \\
+ \lambda \left( \sum_{x_i \in X} ||x_i||^2 + \sum_{y_j \in Y} ||y_j||^2 \right),
\]

where \(x_i (\in X), y_j (\in Y)\) are the latent factor vectors of a user \(i\) and a music item \(j\), \(p_{i,j}(\in \{0, 1\})\) represents the el-
Auxiliary information on music recommendation

One of the approaches for improving the accuracy of recommendation with implicit feedback is to introduce auxiliary information, such as social information [Nanopoulos 09, Symeonidis 08] and context information. In the case of music recommendation, content and context features are often used as auxiliary information. Content features include key, tempo, and signal features such as duration, mode, energy and valence of a song. However, these features are low-level descriptors of music items compared with user preference [Celma 10]. For example, a user may not like a song only because the key of this song is major A or its acousticness is too high. Instead, s/he would be more inclined to play a song that makes him/her feel pleasure. It is difficult to precisely represent these kinds of user preference with content features, which is called a problem of the semantic gap [Celma 10].

To reduce the semantic gap, CNN (convolutional neural network) has been used to extract characteristics affecting user preference directly from audio signals [Oord 13].

Regarding context features of music items, Kaminskas, et al. [Kaminskas 12] have classified context features in the music domain into user-related, environment-related, and multi-media features.

- user-related features: emotional state, demographic information, and activity of a user.
- environment-related features: location, current time, weather, and noise level.
- multi-media features: image of an album, lyrics and review of a song.

Deng, et al. have explored user’s emotions in microblogs service for music recommendation [Deng 15]. Following the hypothesis that users have a higher similarity with each other when they played similar music in the same emotional state, this method utilizes the text of microblogs service to extract users’ emotions at different granularity levels with different sizes of a time window. Users’ emotion when playing a song is utilized to calculate the similarity between users and items. Wang, et al. have enhanced the emotion-aware approach by modeling the relations among user, music, and emotion as an emotion aware graph [Wang 16]. Kaminskas, et al. proposed a method to recommend music items suited for places of interest on the basis of the auto-tagging and the knowledge of the semantic relations among items [Kaminskas 13].

As multi-media information, Pichl, et al. have proposed a pre-filtering method, which divides users into clusters by the name of playlists given by users and recommends music items for each user cluster with nearest neighbor-based CF [Pichl 15]. However, it was found that recommendation accuracy substantially varied among clusters because the size of each cluster is different. To improve the performance of this approach, they have used the cluster label as one of the features of FMs to overcome the drawbacks of dividing user groups in advance by pre-filtering.

Schedl, et al. have introduced multiple types of context information into hybrid music recommendation [Schedl 13]. In addition to content features such as the rhythm of a music item, user contexts including timestamp, location, and genres are used in the process of similarity computation.

As for other kinds of auxiliary information, Schedl, et al. have introduced the concept of user mainstreamness to judge whether users prefer music items that are currently popular or not [Schedl 17b].

3. Proposed Method

Figure 1 shows the flowchart of the proposed method. This paper employs FMs [Rendle 12] to recommend music items, in which users, music items, users’ context, and
\[ \hat{r}(e) := \omega_0 + \sum_{i=1}^{t} \omega_i e_i + \sum_{i=1}^{t} \sum_{j=i+1}^{t} \hat{\omega}_{i,j} e_i e_j, \]  

where \( \hat{r}(e) \) is the predicted rating value for \( e \), and \( e_i \) is the value of \( i \)-th feature of \( e \). Note that this paper sets actual rating \( r(e) \) to be 1 for positive samples, and 0 for negative samples. \( \omega_0 \) is the global bias, and \( \omega_i \) is the weight of \( e_i \). \( \hat{\omega}_{i,j} \) shows the interaction of \( i \)-th and \( j \)-th features, which is represented as \( f \)-dimensional vectors (Eq. (3)). It can capture all pairwise interaction between features. \( v_{i,k} \) is \( k \)-th latent factor of feature \( i \).

\[ \hat{\omega}_{i,j} := \sum_{k=1}^{f} v_{i,k} v_{j,k}. \]  

3.1 Negative sampling

This paper employs alternating least squares [Rendle 11] to learn parameters \( \omega_0, \omega_i \), and \( \hat{\omega}_{i,j} \) of FMs. In the learning process, both positive and negative samples are necessary. However, only positive samples (LEs) could be obtained directly when users play music items. Regarding all music items a user has yet to play as negative samples is a straightforward approach as mentioned above. Although it could be effective in matrix factorization-based methods such as wALS [Hu 08], the number of negative samples becomes huge, which causes a problem of time complexity especially when training FMs. To solve this problem, this paper generates negative samples by selecting a part of music items that a user has yet to play. This paper employs a simple approach of static negative sampling. Three types of negative sampling methods are employed in this paper: Random sampling (RS), Top Discount Popularity Sampling (TDPS), and Priority Popularity Sampling (PPS). Popularity is one of the widely-used information in negative sampling methods [Pan 08, Rendle 14]. The proposed methods use popularity differently from existing studies such as the item-oriented sampling [Pan 08] that regards unpopular items as negative samples.

This paper defines the parameter \( \epsilon \) as the ratio of negative samples to positive samples. In other words, the proposed method selects \( \epsilon \) negative sample(s) for each LE. The effect of setting \( \epsilon \) is investigated in the experiment in Chapter 4.

§ 1 Random Sampling

This method selects negative samples in according with the hypothesis that if a music item is not played by a user in some context, s/he would not be interested in it in that context. When training FMs, negative samples are generated from a LE \( e \) by replacing \( e \)'s music item with a music item randomly selected from those which have been not played with the same context and user as \( e \). That is, the same values of context features as \( e \) is given to the negative sample(s).
§ 2  Top Discount Popularity Sampling

Although selecting negative samples at random is a fair method, it is supposed that the popularity of a music item could influence the result of learning FMs. Therefore, in addition to the same hypothesis as RS, this method adopts another hypothesis that if a music item is popular, but not played by a user, it would be a negative sample for this user. A sampling method based on this hypothesis increases the probability of selecting a part of popular items as negative samples. Probability of a music item $i$ being selected as a negative sample is defined as:

$$P_{TDPS}(i) = \begin{cases} 
\frac{2}{(2-\gamma)n}, & \text{popular items} \\
\frac{1}{(2-\gamma)n}, & \text{unpopular items}
\end{cases},$$  \hspace{1cm} (4)

where $\gamma \in [0, 1]$ is a hyper-parameter to determine whether or not a music item is popular, and $n$ is the number of music items. Music items in the top $(1-\gamma) \times 100$ % of a popularity ranking list are regarded as popular. After selecting a music item for a LE in accordance with the popularity, the context is given in the same manner as RS.

§ 3  Priority Popularity Sampling

The TDPS only focuses on a part of music items and uniformly increases the probability of selecting them as negative samples. On the other hand, a method that can tune the probability of a music item being selected as a negative sample more flexibly might be preferable. To realize this idea, this paper also considers a sampling method, which is similar to the method used in reinforcement learning [Schaul 15]. In the proposed method, TD (temporal difference) error in reinforcement learning is replaced with the popularity of a music item. Probability of a music item $i$ being selected as a negative sample is defined as:

$$P_{PPS}(i) = \frac{q_i^{\delta}}{\sum_k q_k^{\delta}},$$  \hspace{1cm} (5)

where $\delta$ is a hyper-parameter, and $q_i$ is defined as:

$$q_i = 1/poprank_i,$$  \hspace{1cm} (6)

where poprank$_i$ denotes the popularity ranking of $i$. According to PPS, a popular music item has higher priority to be selected as a negative sample.

3.2  Content features of music items

As noted in Section 2.2, using low-level content features of a music item directly is expected to be ineffective. To utilize content features, this paper proposes to use the labels of clusters created from user profile regarding music contents as a feature of FMs. A user profile of a user $u$ is calculated based on the available contents features in a dataset. This paper proposes two types of user profile: user genre profile (UGP) [Schedl 17a] and user content profile (UCP).

The UGP of a genre $j$ in $u$’s play history is defined based on the play counts:

$$UGP_{u,j} = \frac{g_{u,j}}{|M_u|},$$  \hspace{1cm} (7)

where $M_u$ is a set of all music items in $u$’s play history, and $g_{u,j}$ represents the play counts of genre $j$ in $M_u$.

The UCP of a content feature $j$ in $u$’s play history is defined as:

$$UCP_{u,j} = \frac{s_{u,j}}{|M_u|},$$  \hspace{1cm} (8)

where $s_{u,j}$ is the summation of a content feature $j$’s values of all music items in $M_u$. Based on UGP and UCP, a cluster label for a user is obtained by applying arbitrary clustering methods such as k-means.

4.  Experiment

4.1  Outline

The proposed music recommendation method is evaluated by using two different datasets: #nowplaying-rs $^4$ and LFM-1b$^5$. #nowplaying-rs is generated from the tweets with hashtag #nowplaying posted in 2014. It contains the name of music items, artists, and six context features (timestamp, tweet language, user language, time zone, hashtags, and sentiment) of users when playing music items [Poddar 18]. Eleven content features (key, mode, acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, valence, and tempo) of music items are also obtained from Spotify API$^6$. Table 2 shows the description of these contents features. For avoiding the influence of tweet bots and extremely inactive users, we removed the users who listened to less than 10 or larger than 5,000 music items. As timestamp is difficult to be used as a context feature as it is, we generated two features from each timestamp: DoW (day of week) and PoD (period of day). We did not use hashtags and sentiment because of only a few LEs included them. Among the

$^4$ http://dbis-nowplaying.uibk.ac.at/#nowplayingrs
$^5$ http://www.cp.jku.at/datasets/LFM-1b/
$^6$ https://developer.spotify.com/documentation/web-api/
Table 2  Description of content features

| Feature       | Type | Description                                      |
|---------------|------|--------------------------------------------------|
| Key           | int  | Overall key                                      |
| Mode          | int  | Major or minor                                   |
| Acousticness  | float| Perceptual measure of intensity and activity ([0,1]) |
| Danceability  | float|                                   |
| Energy        | float|                                   |
| Instrumentalness | float | Whether a track contains no vocals or not ([0,1]) |
| Liveness      | float| Presence of an audience ([0,1])                   |
| Loudness      | float| Overall loudness (dB)                           |
| Speechiness   | float| Presence of spoken words ([0,1])                  |
| Valence       | float| Musical positiveness ([0,1])                      |
| Tempo         | float| Overall estimated tempo (BPM)                    |

The UCP (Section 3.2) is calculated on the basis of eight content features of music items: instrumentalness, liveness, speechiness, danceability, valence, tempo, acousticness, and energy. Loudness is not used because most of the values are none. We focused on numerical features: categorical features such as key and mode are not used. The number of clusters is set to 5, which is determined by a preliminary experiment. Note that UGP cannot be calculated for this dataset because it does not include genre information.

Schedl generated LFM-1b dataset by collecting LEs from more than 120,000 users of Last.fm from 2012 to 2013 [Schedl 16]. It contains user ID, music ID, artist ID, and four context features of a user (timestamp, nationality, age, and gender). As the density of LFM-1b is higher than #nowplaying-rs significantly, we removed the users who listened to less than 20 or larger than 5,000 music items. The number of UGP clusters is set to 5, which is equal to that of UCP of #nowplaying-rs dataset. UCP cannot be calculated because the content features of music items are not included in this dataset. These two datasets are summarized in Table 3.

This paper employs 5-fold cross-validation with real-life split strategy [Chou 15], which divides a dataset into 5 folds after sorting LEs in ascending order of timestamp. When a fold is used as test data, all precedent folds are used as training data. The 5-fold cross-validation is repeated 5 times. The hyper-parameter of the proposed method, \( f \) in Eq. (3), is set to 15 for #nowplaying-rs and 35 for LFM-1b respectively with hyper-parameter tuning.

As a baseline method, wALS is employed because it is one of the popular music recommendation algorithms. Hyper-parameters of wALS are \( \alpha = 250 \), \( f = 15 \), \( \lambda = 0.05 \) for #nowplaying-rs and \( \alpha = 250 \), \( f = 35 \), \( \lambda = 0.05 \) for LFM-1b, which are determined with hyper-parameter tuning.

This paper considers top-\( N \) recommendation task: a recommendation list containing \( N \) music items that have higher predicted rating values than others is generated for each user. Different from recommendation without considering context, context-aware music recommendation supposes that each LE is selected and played under different context each time. That is, it is supposed that a recommendation list is generated for each LE in the test dataset. Therefore, such evaluation metrics as precision, recall, MAP, and NDCG, which suppose to have multiple relevant items for a query, are not suitable in this paper. As an evaluation metric that can be applied to the situation where only a single relevant item is available for each recommendation list, this paper employs MPR (Mean Percentile Rank).

\[
MPR = \frac{\sum_{u,i} h_{u,i} \text{rank}_{u,i}}{\sum_{u,i} h_{u,i}},
\]

where \( \text{rank}_{u,i} \) represents the percentile-ranking of a music item \( i \) in the recommendation list for a user \( u \), and \( h_{u,i} = 1 \) if \( u \) actually listened to \( i \), otherwise 0. Although MRR (Mean Reciprocal Rank) is also available, it highly evaluates the top two or three items in the recommendation list. Considering the difficulty in context-aware music recommendation, we decided to use MPR because
it is less affected by a small number of highly-ranked items than MRR. To show different properties of the proposed method, this paper also employs RMSE (Root Mean Square Error) as the secondary metric.

$$\text{RMSE} = \sqrt{\frac{\sum_{e \in D_{\text{test}}} (r(e) - \hat{r}(e))^2}{|D_{\text{test}}|}},$$

where $D_{\text{test}}$ is a test data. Smaller MPR/RMSE indicates better performance.

### 4-2 Results

Table 4 compares the MPR of wALS and the proposed method without context/content feature. RS with $\epsilon = 3$ is used to select negative samples for the proposed method. In the case of #nowplaying-rs, significant difference with a significance level of 0.05 was observed between wALS and the proposed method.

Figure 2 and Figure 3 show the effect of $\epsilon$ on MPR for #nowplaying-rs and LFM-1b, respectively. No context/content feature is used in this experiment. In #nowplaying-rs, it was observed that setting $\epsilon$ larger than 1 obtained more accurate results than wALS. A similar trend was observed in LFM-1b except $\epsilon = 1$. For example, in the case of #nowplaying-rs, the computation time when $\epsilon = 2$ and 6 are 688(s) and 2,142(s) per fold in cross-validation, respectively. This means that $\epsilon$ should be set by considering the balance between time complexity and accuracy of recommendation.

Table 5 and Table 6 respectively show the performance of the proposed sampling methods in #nowplaying-rs and LFM-1b. As the size of LFM-1b is much larger than #nowplaying-rs, we tested the parameters ($\gamma$ for TDPS and $\delta$ for PPS) up to 0.2. The result with $\gamma/\delta = 0$ corresponds to RS. A significant difference with a significance level of 0.05 was observed only for MPR between random sampling and TDPS with $\gamma = 0.1$ in #nowplaying-rs. Although further investigation is required, this result implies that TDPS is effective for relatively sparse dataset such as #nowplaying-rs.

Table 7 shows the effect of each context/content feature in #nowplaying-rs. In the experiment, $\epsilon$ is set to 3. In the table, features with boldface indicate that significant difference with a significance level of 0.05 was observed compared with the result with no context/content feature (None) except the last row. A significant difference in RMSE was observed for DoW, PoD, Lang, and UCP. Furthermore, a significant difference in MPR was also observed for Lang and UCP. The result shows the effectiveness of Lang and the proposed UCP cluster label. The last row in Table 7 shows the combination of features, UCP+Lang+PoD+DoW. A significant difference between UCP and UCP+Lang+PoD+DoW was observed in MPR. It was also shown that MPR of UCP, which obtained the best result, was improved by combining other features.

It is assumed that if different music items tend to be played for different values of a specific feature, the feature would be effective for learning FMs. Although UCP and UGP are calculated on the basis of the content features of music items, different values are applied to a music item depending on those who played it. To verify this assumption, this paper defines the distance between the popularity (play count) distributions $D_i$ and $D_j$, which correspond

![Fig. 2 Effect of $\epsilon$ on MPR in #nowplaying-rs](image-url)
Table 7 Performance of context/content features in #nowplaying-rs

| Context/Content | MPR       | RMSE     |
|-----------------|-----------|----------|
| None            | 0.104362  | 0.148141 |
| DoW             | 0.103748  | 0.136077 |
| PoD             | 0.104370  | 0.135286 |
| Lang            | 0.104246  | 0.137477 |
| UCP             | 0.102236  | 0.133388 |
| UCP+Lang+PoD+DoW| 0.099829  | 0.135616 |

Table 8 JS divergence of context/content features in #nowplaying-rs

| Context/Content | # feature values | Max  | Min  | Avg  |
|-----------------|------------------|------|------|------|
| Lang            | 10               | 1.9  | 0.51 | 1.03 |
| DoW             | 7                | 0.12 | 0.09 | 0.10 |
| PoD             | 4                | 0.21 | 0.1  | 0.14 |
| UCP             | 5                | 2.5  | 0.44 | 1.37 |

Table 9 Performance of context/content features in LFM-1b

| Context/Content | MPR       | RMSE     |
|-----------------|-----------|----------|
| None            | 0.091355  | 0.159674 |
| Age             | 0.092033  | 0.158702 |
| Nationality     | 0.091560  | 0.157233 |
| PoD             | 0.091977  | 0.158082 |
| DoW             | 0.092011  | 0.158475 |
| UGP             | 0.091140  | 0.157800 |
| Nationality+UGP | 0.090364  | 0.156623 |

Table 10 JS divergence of context/content features in LFM-1b

| Context/Content | # feature values | Max  | Min  | Avg  |
|-----------------|------------------|------|------|------|
| Nationality     | 9                | 1.2  | 0.61 | 0.91 |
| DoW             | 7                | 0.14 | 0.11 | 0.13 |
| PoD             | 4                | 0.17 | 0.09 | 0.14 |
| Age             | 3                | 0.75 | 0.36 | 0.53 |
| UGP             | 5                | 2.2  | 0.67 | 1.22 |

Fig. 3 Effect of $\epsilon$ on MPR in LFM-1b

to different values $x_i$ and $x_j$ of a feature (e.g. “English” and “Japanese” of a feature “Lang”), based on JS (Jensen-Shannon) divergence.

$$JS(D_i, D_j) = \frac{1}{2} \{KL(D_i || D_j) + KL(D_j || D_i)\},$$  \hspace{1cm} (11)

where $KL(D_i || D_j)$ denotes KL-divergence [Kullback 51]. $D_i, D_j$ are calculated for a feature as follows.

1. Collect $M_{i,j}$, a set of music items that were played relating to both of feature values $x_i$ and $x_j$.
2. Collect $L_i^M (L_j^M)$, a set of LEs about music items in $M_{i,j}$ that relate with $x_i$ ($x_j$).
3. Obtain the popularity distribution $D_i, D_j$ based on the play count of music items in $L_i^M, L_j^M$.

Table 8 shows the maximum, minimum, and average JS divergence between different values for each feature. It is shown that effective features such as Lang and UCP have larger values than DoW and PoD. It was also observed that the value of JS divergence did not relate to the number of feature values.

Table 9 shows the effect of each context/content feature in LFM-1b. The result shows that no significant difference with a significance level of 0.05 was observed when each feature was used independently. However, a significant difference was observed when both Nationality and the proposed UGP cluster label was used. The same tendency as #nowplaying-rs was observed as shown in Table 10: JS divergence of Nationality and UGP are higher than other features, and the value of JS divergence did not relate to the number of feature values.

To examine the effect of UCP in #nowplaying-rs, we trained FMs using a content feature directly as a feature of FMs. Table 11 shows significant difference was not observed between the result with any content feature and the result without using a content feature (None). Combining all of the features (All) is not effective as well. On the other hand, a significant difference with a significance level of 0.05 was observed between None and UCP. This result shows the effectiveness of the UCP as an abstract content feature.

5. Conclusion

This paper proposed a method for recommending music items without explicit feedback. Context and content features are used as auxiliary information to compensate for implicit feedback.

The proposed method employs FMs, in which the context/content information is treated as features. Instead of using content features of music items directly, this paper
employed more abstract features UGP and UCP, which are generated from users’ play histories. As FMs need both positive and negative samples to learn their parameters, three negative sampling methods were proposed: RS, TDDS, and PPS.

The effectiveness of the proposed method and the effect of negative sampling methods were evaluated in terms of MPR with #nowplaying-rs and LFM-1b dataset. The result of the experiment showed that the proposed method outperformed wALS.

Regarding the detailed analysis of the characteristics of the proposed method, it was found that the negative sample ratio should be set by considering the balance between time complexity and the accuracy of recommendation. The effectiveness of each feature was examined, and it was found that a feature is effective when JS divergence of popularity distribution between different feature values is large. As for content features, the UCP and UGP cluster labels were shown to be more effective than using content features directly. The findings of the experiments on the two large music datasets are expected to be valuable to the development of music recommender systems.

In future works, negative sampling methods considering different factors than popularity or more sophisticated approaches such as dynamic[Rendle14]/supervised[Ding19] ones should be considered. Extending the proposed method to playlist recommendation is also one of the challenging research topics.

Acknowledgments

This work was partly supported by JSPS KAKENHI Grant Numbers JP19K22896.

Table 11 Performance of UCP

| Context/Content | MPR     | RMSE     |
|-----------------|---------|----------|
| None            | 0.104362| 0.148141 |
| Instrumentalness| 0.109320| 0.142889 |
| Liveness        | 0.104577| 0.138641 |
| Speechness      | 0.104671| 0.144179 |
| Danceability    | 0.104026| 0.135582 |
| Valence         | 0.104342| 0.135105 |
| Tempo           | 0.193850| 0.209721 |
| Acousticness    | 0.104289| 0.137100 |
| Energy          | 0.106400| 0.139753 |
| All             | 0.223682| 0.212018 |
| UCP             | 0.102236| 0.131457 |

References

[Adomavicius 15] Adomavicius, G., Tuzhilin, A., “Context-aware recommender systems,” F. Ricci, L. Rokach, B. Shapira eds., Recommender Systems Handbook, pp. 191–226, Springer (2015).
[Celma 10] Celma, O., “Music recommendation,” Music Recommendation and Discovery, pp. 43–85, Springer (2010).
[Chou 15] Chou, S., Yang, Y., Lin, Y., “Evaluating music recommendation in a real-world setting: On data splitting and evaluation metrics,” 2015 IEEE International Conference on Multimedia and Expo, pp. 1–6 (2015).
[Deng 15] Deng, S., Wang, D., Li, X., Xu, G., “Exploring user emotion in microblogs for music recommendation,” Expert Systems with Applications, Vol. 42, No. 23, pp. 9284–9293 (2015).
[Ding 19] Ding, J., Quan, Y., He, X., Li, Y., Lin, D., “Reinforced negative sampling for recommendation with exposure data,” 78th International Joint Conference on Artificial Intelligence (IJCAI-19), pp. 2230–2236 (2019).
[Hu 08] Hu, Y., Koren, Y., Volinsky, C., “Collaborative filtering for implicit feedback datasets,” 8th IEEE International Conference on Data Mining, pp. 263–272 (2008).
[Johnson 14] Johnson, C., C., “Logistic matrix factorization for implicit feedback data,” NIPS2014 Workshop on Machine Learning and Matrix Computations (2014).
[Kaminskas 12] Kaminskas, M., Ricci, F., “Contextual music information retrieval and recommendation: State of the art and challenges,” Computer Science Review, Vol. 6, pp. 89–119 (2012).
[Kaminskas 13] Kaminskas, M., Ricci, F., Schedd, M., “Location-aware music recommendation using auto-tagging and hybrid matching,” 7th ACM Conference on Recommender Systems, pp. 17–24 (2013).
[Kullback 51] Kullback, S., Leibler, R., A., “On information and sufficiency,” The Annals of Mathematical Statistics, Vol. 22, No. 1, pp. 79–86 (1951).
[Nanopoulos 09] Nanopoulos, A., Rafalidis, D., Symeonidis, P., Nanopoulos, Yi., “Musibox: Personalized music recommendation based on cubic analysis of social tags,” IEEE Transactions on Audio, Speech, and Language Processing, Vol. 18, No. 2, pp. 407–412 (2009).
[Oard 98] Oard, D., W., Kim, J., “Implicit feedback for recommender systems,” AAAI Workshop on Recommender Systems, Vol. 83, pp. 81–83 (1998).
[Oard 13] van den Oord, A., Dieleman, S., Schrauwzen, B., “Deep content-based music recommendation,” Advances in Neural Information Processing Systems, pp. 2643–2651 (2013).
[Pan 08] Pan, R., Zhou, Y., Cai, B., Liu, N., N., Lukose, R., Scholz, M., Yang, Q., “One-class collaborative filtering,” 8th IEEE International Conference on Data Mining, pp. 502–511 (2008).
[Pichl 15] Pichl, M., Zangerle, E., Specht, G., “Towards a context-aware music recommendation approach: What is hidden in the playlist name,” IEEE International Conference on Data Mining Workshop, pp. 1360–1365 (2015).
[Poddar 18] Poddar, A., Zangerle, E., Yang, Y., “#nowplaying-rs: A new benchmark dataset for building context-aware music recommender systems,” 15th Sound and Music Computing Conference (2018).
[Rendle 09] Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L., “BPR: Bayesian personalized ranking from implicit feedback,” 25th Conference on Uncertainty in Artificial Intelligence, pp. 452–461 (2009).
[Rendle 11] Rendle, S., Gantner, Z., Freudenthaler, C., Schmidt-Thieme, L., “Fast context-aware recommendations with factorization machines,” Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 635–644 (2011).
[Rendle 12] Rendle, S., “Factorization machines with libfm,” ACM Transactions on Intelligent Systems and Technology, Vol. 3, No. 3, Article No. 57, pp. 1–22 (2012).
[Rendle 14] Rendle, S., Freudenthaler, C., “Improving pairwise learning for item recommendation from implicit feedback,” 7th ACM International Conference on Web Search and Data Mining
(WSDM’14), pp. 273–282 (2014).

[Salakhutdinov 07] Salakhutdinov, R., Mnih, A., “Probabilistic matrix factorization,” Advances in Neural Information Processing Systems, pp. 1257–1264 (2007).

[Schaal 15] Schaal, T., Quan, J., Antonoglou, I., Silver, D., “Prioritized experience replay,” arXiv preprint arXiv:1511.05952 (2015).

[Schedl 13] Schedl, M., Schnitzer, D., “Hybrid retrieval approaches to geospatial music recommendation,” 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 793–796 (2013).

[Schedl 16] Schedl, M., “The lfms-1b dataset for music retrieval and recommendation,” Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, pp. 103–110 (2016).

[Schedl 17a] Schedl, M., Ferwerda, B., “Large-scale analysis of group-specific music genre taste from collaborative tags,” 2017 IEEE International Symposium on Multimedia, pp. 479–482 (2017).

[Schedl 17b] Schedl, M., Bauer, C., “Distance- and rank-based music mainstreamness measurement,” 25th Conference on User Modeling, Adaptation and Personalization, pp. 364–367 (2017).

[Symeonidis 08] Symeonidis, P., Ruxanda, M., Nanopoulos, A., Manolopoulos, Y., “Ternary semantic analysis of social tags for personalized music recommendation,” International Society for Music Information Retrieval, Vol. 8, pp. 219–224 (2008).

[Wang 16] Wang, D., Deng, S., Xu, G., “Gemrec: A graph-based emotion-aware music recommendation approach,” International Conference on Web Information Systems Engineering, pp. 92–106 (2016).

[Zhang 20] Zhang, J., Takama, Y., “Proposal of context-aware music recommender system using negative sampling,” Ohnawa, Y., Yada, K., Ito, T., Takama, Y., Sato-Shimokawara, E., Abe, A., Mori, J., Matsumura, N., eds., Advances in Artificial Intelligence: Selected Papers from JSAI2019, pp. 114–125, Springer (2020).

[Zhang 13] Zhang, W., Chen, T., Wang, J., Yu, Y., “Optimizing top-N collaborative filtering via dynamic negative item sampling,” Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 785–788 (2013).

Received May 25, 2020.

Author’s Profile

Takama, Yasufumi (Member)
He received his Dr. Eng. degree from The University of Tokyo, Tokyo, Japan in 1999. He was a JSPS (Japan Society for the Promotion of Science) Research Fellow from 1997 to 1999. From 1999 to 2002 he was a Research Associate at Tokyo Institute of Technology in Japan. From 2002 to 2005, he was an Associate Professor at Tokyo Metropolitan Institute of Technology, Tokyo, Japan. From 2005 to 2013, he was an Associate Professor at Tokyo Metropolitan University, Tokyo, Japan. Since 2014, he has been a Professor at Tokyo Metropolitan University. He also participated in PREST (Precursory Research for Embryonic Science and Technology), JST (Japan Science and Technology Corporation) from 2000 to 2003. His current research interest includes Web intelligence, information visualization, recommendation, and intelligent interaction. He is a member of IEEE, ACM, IEICE (Institute of Electronics, Information and Communication Engineers), and IPSJ (Information Processing Society of Japan).

Zhang, Jing-cheng
He received his M.E. degree from Tokyo Metropolitan University, Tokyo, Japan in 2019. His research interest includes music recommendation.

Shibata, Hiroki (Member)
He received his Dr. Eng. degree from Tokyo Metropolitan University, Japan in 2019. Since April 2019, he has been an Assistant Professor at Tokyo Metropolitan University. His current research interest includes data mining, route recommendation, probability model, optimization, and machine learning theory. He is a member of IEEE, ACM, and IEICE (Institute of Electronics, Information and Communication Engineers).