Deep Reinforcement Learning-based Music Recommendation with Knowledge Graph Using Acoustic Features

Keigo Sakurai††, Ren Togo†††, Takahiro Ogawa (member)††††, Miki Haseyama (fellow)†††

Abstract In this study, we propose a new deep reinforcement learning-based music recommendation method with knowledge graphs. With the rapid development of Web services, music-related content posted on platforms, such as YouTube, is increasing dramatically. Conventional recommendation methods based on knowledge graphs have struggled with the cold-start problem caused by a lack of user preference information. The proposed method can solve this problem by introducing acoustic feature edges in the constructed knowledge graph. Furthermore, we realize efficient search using a deep reinforcement learning algorithm on a dense knowledge graph introducing acoustic feature-based edges. The proposed method can make appropriate recommendations even with a small amount of user preference information by learning the optimal action of the agent. We confirm the effectiveness of the proposed method by comparing our method with several conventional and state-of-the-art recommendation methods.

Key words: Music recommendation system, knowledge graph, reinforcement learning, cold-start problem, graph analysis, graph embedding.

1. Introduction

With the rapid growth of the Internet and mobile technologies, we can easily access an enormous number of digital contents on the Web [1]. Although we have the opportunity to access a wide variety of content, it has become challenging to determine contents that we are interested in [2,3]. Thus, recommendation systems have become one of the important tools to assist users in searching, sorting, and filtering many digital contents [4].

Currently, it is not an exaggeration to say that almost every service providing content to users is equipped with recommendation systems [5]. Among them, music is one of the most relevant contents for recommendation technology. For example, on YouTube*, it has been reported that 70% of users watch the videos presented by the recommendation system [6]. Moreover, different categories of videos are posted on YouTube every day, and one of the most popular categories is music. Further, 23% of the posted videos and 86% of the retrieved queries are related to music [7,8]. The revenue of music streaming services accounts for 51% among the entire music industry, providing physical formats and digital downloads, amounting to a total of US$7.5 billion. Digital music sales have been more prevalent than physical sales. Since we are always surrounded by music and music-related content, satisfying the demand based on the music recommendation system will contribute to the comfort of our daily life.

The music-related recommendation methods can provide music content based on the relationships between a target user and users with similar tastes [9]. One of the representative music recommendation methods is the knowledge graph-based method; it consists of a semantic network that describes entities corresponding to recommendation technology. For example, on YouTube*, it has been reported that 70% of users watch the videos presented by the recommendation system [6]. Moreover, different categories of videos are posted on YouTube every day, and one of the most popular categories is music. Further, 23% of the posted videos and 86% of the retrieved queries are related to music [7,8]. The revenue of music streaming services accounts for 51% among the entire music industry, providing physical formats and digital downloads, amounting to a total of US$7.5 billion. Digital music sales have been more prevalent than physical sales. Since we are always surrounded by music and music-related content, satisfying the demand based on the music recommendation system will contribute to the comfort of our daily life.

The music-related recommendation methods can provide music content based on the relationships between a target user and users with similar tastes [9]. One of the representative music recommendation methods is the knowledge graph-based method; it consists of a semantic network that describes entities corresponding to users and contents to consider their high-order relationships [10,11]. The advantage of knowledge graph-based methods is their high explainability. When the recommendation path of the knowledge graph is \{User X → Song A → User Y → Song B\}, it means that the system recommends Song B to User X since User Y who listened to Song A in the past also listened to Song B.

This explainability is a crucial element to convince the recommendation and helps improve the users’ sat-
satisfaction of recommendation [12].

Despite the improvement of knowledge graph-based methods using user preference information, the cold-start problem is still one of the remaining key challenges [13]. The cold-start problem means that the recommendation systems do not work well due to the lack of user preference information [14]. For example, music streaming services can rarely recommend new users, who have just subscribed to the service, since the system has little information about their preferences. To address this problem, integrating content and user preference information in a hybrid way has been reported as one of the effective solutions [15–19]. Hybrid methods can solve the problem by supplementing insufficient user preference information with content-specific information, i.e., acoustic features. Introducing content information into the conventional knowledge graph-based methods is expected to solve the cold-start problem and achieve high explainability.

In this study, we propose a deep reinforcement learning-based music recommendation with a knowledge graph using acoustic features. The proposed method can make an accurate recommendation from a small number of users’ preference information with high explainability. We need to solve the following newly arising problem to use content information and users’ preference information in the knowledge graph-based recommendation. Since the density of the knowledge graph constructed from users’ listening histories and acoustic similarities becomes high, it becomes challenging to determine the optimal recommendation path from the target user to recommended songs over the graph.

To deal with the newly arising problem, we introduce the concept of reinforcement learning for the recommendation task [20]. Reinforcement learning is a machine learning method for obtaining optimal actions of an agent in a given task based on a reward function [21]. We design an intelligent recommendation agent that can conduct explicit reasoning over a knowledge graph. An agent can explore nodes on the graph regarding the knowledge graph as an environment for reinforcement learning. Additionally, the optimal action of the agent can be learned based on the reward designed by the relationships of the node features. In this way, we apply a reinforcement learning algorithm to fit the task of music recommendation. We can perform an efficient search even on the dense knowledge graph by making recommendations using the optimized path on the knowledge graph. Furthermore, high explainability can be maintained by following the optimal path of action of the agent. Consequently, a successful music recommendation with high explainability becomes feasible.

The contribution of this study is summarized as follows:

(i) We construct an advanced knowledge graph by adding acoustic feature edges to solve the cold-start problem.

(ii) We achieve improved explainability by explicitly defining rewards in our reinforcement learning-based recommendation.

The remainder of the paper is organized as follows: In Section 2, we present related works. Section 3 explains the proposed music recommendation method based on deep reinforcement learning and a knowledge graph using acoustic features. Section 4 presents the experimental results using a real-world music listening history dataset. Finally, Section 5 presents the conclusion.

2. RELATED WORKS

In this section, we present related studies on music and explainable recommendations to clarify the novelty and contribution of our study.

2.1 Music Recommendation

Music recommendation methods are categorized into three types: collaborative filtering, content-based filtering, and the hybrid methods based on the prior two methods. Collaborative filtering methods recommend songs by employing user preference information of other like-minded users [10, 22, 23]. Although collaborative filtering methods can be implemented easily and provide highly accurate recommendation results to users, it tends to fall into the cold-start problem. However, content-based filtering methods try recommending songs based on similarities between the content features [24–26]. These methods solve the cold-start problem and recommend new songs to other users who have never listened to them yet. However, it is not easy to understand why the recommended songs are provided since the recommendation is based on acoustic features in the content-based filtering methods. Based on the advantages and disadvantages of these two methods, the hybrid methods derived from them have been proposed. The hybrid methods enable the consideration of user preference and content-specific information, i.e., acoustic features [16]. Therefore, we focus on the hy-
brid method to solve the cold-start problem and achieve high explainability.

2.2 Knowledge Graph-based Recommendation

Knowledge graphs have been investigated as tools for achieving accurate and explainable recommendation [27]. More semantic information can be introduced to the recommendation using knowledge graphs, and it becomes feasible to improve its performance. Furthermore, explainable recommendation attempts to provide suitable content to users along with its reason [28]. Since knowledge graphs have nodes and edges representing relationships with semantic descriptions, the knowledge graph-based recommendation can explain its reason by following paths leading to the recommended content through the graph. Zhang et al. introduced various heterogeneous features, such as textual and visual features of items, into the knowledge graph to improve recommendation accuracy [29]. Wang et al. proposed an RNN-based model to infer the knowledge graph for recommendation [30]. However, it is not practical because it requires enumeration of all possible paths between each user-item pair to train and predict the model. In order to deal practically with the knowledge graph, they introduced a graph attention network into the knowledge graph to make recommendation based on the importance of the path between users and contents [31]. They also proposed a recommendation method focusing on exploring user intents behind user-item interactions by using auxiliary item knowledge to improve the performance of the recommendation [32].

In recent years, reinforcement learning-based methods have been actively studied in the field of recommendation. Zhou et al. leveraged a prior knowledge in a knowledge graph to achieve high sample efficiency in a reinforcement learning-based interactive recommendation system [33]. Xian et al. attempted the knowledge graph reasoning for recommendation by employing the soft reward function based on knowledge graph embeddings [34]. Song et al. employed a hard reward function based on user interaction information in addition to a soft reward function to achieve the improvement of the recommendation accuracy [35]. In this way, by focusing on the reward functions of reinforcement learning, it is possible to realize recommendation different from that based on supervised learning.

While reinforcement learning-based recommendation is promising, there are some issues to be improved. For example, as mentioned in the above methods, there still remain the following two problems to be solved. First, although they employed soft reward functions, they still did not make explicit reference to the clear significance and effectiveness of this reward function. Reward functions have a great impact on the performance of reinforcement learning algorithms, and ideally, the rationale for designing the reward function must be clear. Secondly, although their methods perform well under the assumption that there are sufficient metadata and user interaction, its performance tends to be degraded by the cold-start problem. In the case of music recommendation, which is the target of this paper, there exist cases where it is difficult to make recommendation under such an environment. Therefore, we attempt to realize highly accurate and highly explainable music recommendation by clearly defining the reward function for recommending music with appropriate reasons in this paper.

3. Our Music Recommendation Method

Figure 1 presents an overview of the proposed method. Our method consists of the following three phases: knowledge graph construction, training of the agent on the knowledge graph, and recommendation of songs based on the trained policy. In the first phase, we construct a knowledge graph using acoustic features of songs in addition to users’ listening histories and artist information of the songs. In the second phase, we train an agent to investigate the constructed knowledge graph based on the deep reinforcement learning algorithm. The music recommendation is realized in the third phase through the obtained agent policy and beam search algorithm.

3.1 Knowledge Graph Construction

First, we construct a knowledge graph $G$, which is newly introduced in the proposed method. Note that the nodes of the knowledge graph correspond to users, artists, and songs. The edges are defined from listening histories, artist information, and acoustic feature similarities. The knowledge graph $G$ having a node set $\mathcal{N}$ and an edge set $\mathcal{E}$ is defined as $G = \{ (n_{\text{head}}, e, n_{\text{tail}}) \mid n_{\text{head}}, n_{\text{tail}} \in \mathcal{N}, e \in \mathcal{E} \}$, where $n_{\text{head}}$ is a head node, $n_{\text{tail}}$ is a tail node and $e$ is an edge connecting $n_{\text{head}}$ with $n_{\text{tail}}$. In the proposed method, these nodes consist of users $u_i$ ($i = 1, 2, ..., N_u$; $N_u$ being the number of users), songs $m_j$ ($j = 1, 2, ..., N_m$; $N_m$ is the number of songs), and artists $a_k$ ($k = 1, 2, ..., N_a$; $N_a$ is the number of artists). Next, we define the directed edge $e_{(u_i, m_j)}$ connecting user $u_i$ to song $m_j$ that user $u_i$ listened to.
previously. Furthermore, the directed edge $e_{i(m_j, a_k)}$ connecting artist $a_k$ to song $m_j$ that artist $a_k$ created is defined. By using a directed graph, we can understand the hierarchical relationships of heterogeneous entities. In this way, we can express the relationships of entities visually and computationally on the knowledge graph, such as “[user $u_i$] listen to [song $m_j$]” expressed as “$u_i \xrightarrow{e_{i(m_j, a_k)}} m_j$”, which forms the basis of explanations for recommendation. As can be seen from the various explainable recommendation studies [30,31,34], the use of directed graphs is a common way to understand the hierarchical structure of entities.

In the proposed method, we newly define edges between two different songs based on their acoustic features with the above two kinds of edges. Specifically, we extract $M$-dimensional acoustic features $f_j$ from each song $m_j$. When the cosine similarity $w_{j_1,j_2} (j_1, j_2 \in \{1, 2, ..., N_m, j_1 \neq j_2\})$ between $f_{j_1}$ and $f_{j_2}$ is greater than a threshold $\xi$, we define a new edge defined as $e_{(m_{j_1}, m_{j_2})}$ between nodes of $m_{j_1}$ and $m_{j_2}$. In this way, we construct the new knowledge graph considering users’ listening histories, artist information, and acoustic features. We can enhance the density of the knowledge graph and supplement the lack of users’ preference information by introducing edges based on the cosine similarity of acoustic features.

### 3.2 Agent Training on Knowledge Graph

An overview of the agent-training phase is shown on the left side of Fig. 1. We train an agent to explore a target user node to a song node based on a deep reinforcement learning algorithm on the knowledge graph $G$ constructed in the previous subsection. A deep reinforcement learning strategy learns the agent policy to maximize the reward by defining the environment, state, action, and reward function. We define each of them for the music recommendation task as follows:

**Environment.** The environment is the knowledge graph $G$ generated in the previous subsection.

**State.** The state that an agent transitioned $t (= 0, 1, ..., T$: $T$ being the number of steps to end the exploration) steps is represented as $s_t = (u_i, n_t, b_t)$. Here, $u_i$ is the user node that is the initial position of the agent, $n_t$ denotes the nodes that the agent reached after step $t$, and $b_t$ denotes the combination of all nodes and edges that the agent passed before step $t$ (i.e., $b_t=(n_0, e_{(m_0, n_1)}, ..., n_{t-1}, e_{(n_{t-1}, n_t)})$, where $e_{(n_{t-1}, n_t)}$ denotes an edge connecting $n_{t-1}$ and $n_t$).

**Action.** The action $b_t$ is the selection of the node that the agent transitions from $n_t$ at step $t$. We define candidates of the action $b_t$ as all possible nodes connecting $n_t$ excluding the nodes that the agent passed before step $t$.

**Reward.** Conventional knowledge graph and reinforcement learning based recommendation methods have designed soft reward functions, however, it is still unclear reference to the significance and effectiveness of these reward functions. In order to clearly express “appropriateness as an explanation of recommendation” in a reward function, we utilize the additive compositional property of the knowledge graph embedding method.
and the cosine similarity of the calculated embedded features.

As preparation for setting up the reward for learning the optimal agent policy, we vectorize the knowledge graph $G$ constructed in the previous subsection using a knowledge graph-embedding technique. Figure 2 shows an overview of calculating rewards. Motivated by TransE [36], one of the major graph-embedding methods, we embed the knowledge graph $G$ to the vector space. We learn a vector representation of the knowledge graph triplet $(n_{\text{head}}, e, n_{\text{tail}})$ to satisfy $n_{\text{head}} + e = n_{\text{tail}}$, where $n_{\text{head}}, e$ and $n_{\text{tail}}$ are vector representations of $n_{\text{head}}, e$ and $n_{\text{tail}}$, respectively.

Here, we define $E_{ui,n_j}$ as a set of edges in the agent-exploring path from the user node $u_i$ at the initial position and the node $n_j$ the agent reached step $t$. Additionally, we define the composite vector representations of the forward and backward reactions of the edges in $E_{ui,n_j}$ as $e_{fw}(u_i,n_j)$ and $e_{bw}(u_i,n_j)$, respectively. Then, we define the reward $R_t$ as follows:

$$R_t = \text{cossim}(u_i, n_j, m_j)$$

Fig. 2 Overview of the calculation method of the reward $R_t$. When the agent explores the knowledge graph $G$ from the user node $u_i$ to the song node $n_j$, the reward $R_t$ is defined as the cosine similarity between $u_i + e_{(u_i, n_j)} + e_{(n_j, m_j)}$ and $e_{(m_j, u_i)} + m_j$.

We assume that the appropriateness of the reason for recommending $m_2$ to $u_1$ can be expressed clearly by the cosine similarity between $e_{fw}(u_i, n_j)$ and $e_{bw}(u_i, n_j)$. If the cosine similarity is high, it means that the recommendation is natural and appropriate, and if it is low, it means that the recommendation is unnatural and inappropriate. To the best of our knowledge, there is no knowledge graph and reinforcement learning based recommendation method that uses the cosine similarity of embedding features by TransE as a reward.

**Optimization.** We calculate the agent policy $\pi_\theta(b_t | s_t)$ that outputs the agent action $b_t$ by inputting the state vector $s_t$, which is the vector representation of the state $s_t$ defined as follows:

$$\pi_\theta(b_t | s_t) = F_{\text{drop}}(\sigma(F_{\text{drop}}(\sigma(s_t)W_1))W_2))W_3, \quad (4)$$
$$\theta = (W_1, W_2, W_3), \quad (5)$$

where $W_1$, $W_2$ and $W_3$ are the parameters of the policy $\pi_\theta(b_t | s_t)$, $s_t$ is the concatenation of the TransE embeddings of $u_i$, $n_j$ and $b_t$, $\sigma(\cdot)$ is an exponential linear unit, which is one of the popular nonlinear activation functions, and $F_{\text{drop}}(\cdot)$ is the Dropout model proposed in [37]. The goal of the optimization is to obtain the set of parameters $\theta$ of the policy $\pi_\theta(b_t | s_t)$ to maximize the objective function $J(\theta)$ as follows:

$$J(\theta) = E_{\pi_\theta}[\pi_\theta(b_t | s_t)G(s_t)], \quad (6)$$
$$G(s_t) = \sum_{t'=t}^{T-1} \gamma^{t'-t}R_{t'+1}, \quad (7)$$

where $E[\cdot]$ is the expected value and $\gamma$ is the discount rate of the reward. The optimization of the objective function $J(\theta)$ is operated by the REINFORCE algorithm [20]. To obtain the set of parameters $\theta$, where $J(\theta)$ is maximized, we learn to minimize the policy gradient as follows:

$$\nabla_\theta J(\theta) = E_{\pi_\theta}[\nabla_\theta \log \pi_\theta(b_t | s_t)G(s_t)]. \quad (8)$$

Although it is not easy to directly determine the set of parameters $\theta$ that maximizes the objective function $J(\theta)$, the REINFORCE algorithm enables us to engage in the optimization analytically.

**3.3 Recommendation of Songs Based on Policy**

An overview of the recommendation phase is shown on the right side of Fig. 1. Based on the trained policy $\pi_\theta(b_t | s_t)$ obtained in the previous subsection and the beam search, we determine the songs and their paths to recommend to the target user. Figure 3 shows an
overview of the beam search algorithm. The beam search is a heuristic search algorithm that stores top-$K$ nodes with high scoring at each step [38]. First, we determine the target user $u_t$ and the number of steps $T$. Next, we make the agent explore the knowledge graph $G$ from the node of the target user $u_t$ based on the beam search algorithm that selects top-$K$ nodes with high transition probability output through the policy $\pi(d(b_t | s_t))$ at each time step. Finally, we recommend songs to the target user in order of the highest reward of paths between the target user and candidate song nodes that the agent reached step $T$.

4. EXPERIMENTS

In this section, we evaluate the effectiveness of the proposed method by comparing our method with popular and state-of-the-art recommendation methods. We present experimental settings in subsection 4.1 and results in subsection 4.2.

4.1 Settings

In this experiment, we used a listening history database of 1,536 users provided by The Music Listening Histories Dataset [39]. We obtained the listening history of each user for the past five years and defined the top 50 songs with the highest number of times listened to as the songs the user listens to. For each song, we obtained the artist information from MusicBrainz \( ^* \). Furthermore, we used six-dimensional acoustic features (danceability, energy, speechiness, acousticness, valence and instrumentalness) from Spotify API \( ^* \). Following [40], we can characterize our dataset songs finely by using these six features. It should be noted that we did not use “mode” because it is a binary value. Instead, we used the float value “instrumentalness”, as utilized in [41], to take into account how much singing voice is included in the song. By using these acoustic features, we can achieve music recommendation that takes into account the user’s preferences for various aspects of songs. For example, “speechiness” is the measure of the presence of spoken words in a track, and we can consider the preferences of users who place importance on lyrics. “Acousticness” and “instrumentalness” denote whether the track is acoustic and whether a track contains no vocals, respectively. By using them, we can consider the preferences of users who place importance on musical instruments. Furthermore, “danceability” indicates how suitable a track is for dancing, and “valence” and “energy” respectively indicate musical positiveness and negativeness. They help recommend songs that are appropriate for the user’s mood. By integrating these six values, we can prepare acoustic features that reflect the multifaceted characteristics of songs. The threshold $\xi$ for the cosine similarity was experimentally set to 0.999. In this way, we used 38,526 songs; 1,500 users; and 6,975 artists as the dataset. In the agent-training phase, we set the number of agent steps $T$ to 4, and the discount rate $\gamma$ to 0.99. In the recommendation phase, we set the sampling size of the beam search $K$ to 4.

First, we evaluated the effectiveness of introducing acoustic feature-based edges for the cold-start problem at the recommendable user rate. Graph-based methods often cannot make reasonable recommendations in the cold-start problem setting due to the lack of edges for songs. To overcome this problem, we introduced acoustic feature-based edges in the proposed method. We defined the rate of users who are recommended more than ten songs as the recommendable user rate and used it when edges corresponding to songs in the bottom $\alpha (=90, 80, 70, 60, 50) \%$ of listening counts of each user are removed. Thus, $\alpha$ represents the ratio of the removed edges of the user’s listening history. Then, when $\alpha$ becomes larger, the cold start problem becomes severe. This approach is a common evaluation metric for the cold-start problem setting, which has been used in [42] and elsewhere. A high value of the metric means that the cold-start problem is mitigated. We compared the proposed method (PM) with a method without acoustic features (PM - AF) and a method without acoustic features and artist information (PM - AF - AI) to confirm the effectiveness of our newly introduced architectures.

We also verified the recommendation performance of
the proposed method in the cold-start problem setting. We used precision, recall, normalized discounted cumulative gain (nDCG), and hit rate as evaluation metrics, which are typical metrics in recommendation methods. In this evaluation, we used the knowledge graph in which the edges of the user’s listening history were removed by α (=90, 80, 70, 60, 50) [%]. By removing most edges of the user’s listening history, we generated the environment of the cold-start problem setting. We recommended 10 songs for the defective knowledge graph and evaluated whether the songs corresponding to the removed edges were recommended. We compared the proposed method with the following four existing methods, including some state-of-the-art recommendation algorithms.

CM1 [31]
Knowledge graph attention network: a state-of-the-art knowledge graph-based method using a sequential recurrent neural network model.

CM2 [43]
Collaborative filtering over knowledge graph model: a collaborative filtering-based method using knowledge graphs and their embeddings based on TransE.

CM3 [44]
Bayesian personalized ranking based on matrix factorization: a popular pairwise ranking method that models the implicit feedback data.

CM4 [29]
Collaborative knowledge-based embedding model: a knowledge base embedding framework that learns the latent representations in collaborative filtering and items’ semantic representations from the knowledge base.

CM5 [19]
A hybrid music recommendation model based on multilayer neural networks via prediction of semantic tags from acoustic features.

Since the method in [30] requires enumeration of all paths and is not suitable for the task of recommendation, we adopted the method in [31], which is a more recent method than [30]. This point is also mentioned in [34,45]. Note that in order to evaluate the proposed method fairly, each comparison method also used acoustic features.

4.2 Results
Table 1 presents the recommendation user rate. Table 1 shows that the values of PM are higher than those of PM - AF and PM - AF - AI in all cases. The PM increases the number of song nodes that have edges by introducing acoustic features. Moreover, the number of recommendable users of each α was PM>PM - AF>PM - AF - AI. These results show that increasing the number of edges based on acoustic features mitigates the cold-start problem.

Next, we show the results of precision, recall, nDCG, and hit rate for each α (= 90, 80, 70, 60 and 50) [%] in Fig. 4. We can see that PM showed higher precision, recall, and hit rate than those of all conventional methods, including the state-of-the-art methods for all α. The performance of PM is maximized at α=70 in all evaluation metrics, except nDCG. This is because when α becomes small and the knowledge graph becomes dense, the number of candidate songs increases, and the effectiveness of the user of acoustic features decreases.

In contrast to other evaluation metrics, PM’s nDCG shows a high value at α=90. Note that nDCG is a metric for rank prediction; it becomes a high value when a correct song is recommended to the top and a low value when it is recommended to the bottom. Since most edges are removed at α=90, there is a large number of recommendation candidates. The PM can recommend candidates for the top position by exploiting acoustic feature-based edges as cues. However, PM’s nDCG is inferior to CM3 and CM4 at α=60 and 70. However, PM’s nDCG is inferior to CM3 and CM4 at α=70, to CM3, CM4 and CM5 at α=60 and to CM1, CM3, CM4 and CM5 at α=50. This indicates that although PM is superior in determining correct songs by exploring the knowledge graph, there is room for improvement in determining the recommendation rank.

Since the recommendation ranking is based on the reward of reinforcement learning, its algorithm needs further improvement. Furthermore, for Precision, Recall and Hit Rate, PM performs equal to or less than CM3, CM4 and CM5 at α = 50. This may be due to the fact that acoustic feature edges interfere with the search for edges in the user’s listening history. However, this is not a surprising result, since a small value of α is not the setting of the cold start problem. In future work, we will build a recommendation model for both the cold

| Table 1 | Recommendable user rate [%] |
|---------|-----------------------------|
| α =90   | 18.5 64.5 90.1 99.9 99.9    |
| α =80   | 12.0 62.3 81.6 88.0 96.0    |
| α =70   | 10.4 43.0 66.9 76.4 88.4    |
| α =60   | 10.4 43.0 66.9 76.4 88.4    |
| α =50   | 10.4 43.0 66.9 76.4 88.4    |

ITE Trans. on MTA Vol. 10, No. 1 (2022)
start and non-cold start problems by combining our recommendation model with other existing models. The proposed method can explain the reason why the song was recommended following a path in the knowledge graph $G$. Therefore, we conclude that our method achieves promising recommendation results and efficiently determines paths for the explainable recommendations.

We confirm the advantages of the proposed recommendation method through several experiments from different perspectives. The proposed method can work well in the cold-start problem setting by introducing acoustic feature-based edges to the knowledge graph. In the future study, we will exploit more metadata of music or music-related social data and improve the reinforcement learning algorithm to achieve more accurate and sophisticated recommendations.

5. CONCLUSION

We have proposed a new music recommendation method based on reinforcement learning using a knowledge graph. The proposed method can mitigate the cold-start problem and provide highly accurate recommendations by introducing the acoustic features of songs into the knowledge graph. The experimental results confirm that the proposed method outperforms existing methods, including state-of-the-art methods. In our experiments, we used all acoustic features normalized but not weighted. Therefore, users’ specific preferences for the song have not been considered. In future work, we will change the weights of these features according to users preferences for songs.

Acknowledgment

This work was partly supported by JSPS KAKENHI Grant Numbers JP17H01744.

References

1) D. Wang, S. Deng and G. Xu, “Sequence-based context-aware music recommendation,” Information Retrieval Journal, vol. 21, no. 2, pp. 230–252 (2018)
2) D. Bawden and L. Robinson, “The dark side of information: overload, anxiety and other paradoxes and pathologies,” Journal of Information Science, vol. 35, no. 2, pp. 180–191 (2009)
3) G. Haubl and V. Trifts, “Consumer decision making in online shopping environments: The effects of interactive decision aids,” Marketing Science, vol. 19, no. 1, pp. 4–21 (2000)
4) M. Montaner, B. López and J. L. De La Rosa, “A taxonomy of recommender agents on the internet,” Artificial Intelligence Review, vol. 19, no. 4, pp. 285–330 (2003)
5) B. D. Reddy, L. Kumar and N. Nelatur, “A review on datasets and tools in the research of recommender systems,” Smart Technologies in Data Science and Communication, pp. 59–70 (2020)
6) C. Stöcker and M. Preus, “Riding the wave of misclassification: How we end up with extreme YouTube content,” in Proceedings of the 22nd International Conference on Human-Computer Interaction, pp. 359–375 (2020)
7) X. Cheng, C. Dale and J. Liu, “Understanding the characteristics of internet short video sharing: YouTube as a case study,” arXiv preprint arXiv:0707.3670, (2007)
8) L. A. Liikkainen and A. Salovaara, “Music on YouTube: User engagement with traditional, user-appropriated and derivative videos,” Computers in Human Behavior, vol. 50, pp. 108–124 (2015)
9) X. Su and T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” Advances in Artificial Intelligence, vol. 2009, pp. 1–19 (2009)
10) R. Mu and X. Zeng, “Collaborative filtering recommendation algorithm based on knowledge graph,” Mathematical Problems in Engineering, vol. 2018, pp. 1–11 (2018)
11) X. Chai, “Diagnosis method of thyroid disease combining knowledge graph and deep learning,” IEEE Access, vol. 8, pp. 149787–149795 (2020)
12) W. Ma, M. Zhang, Y. Cao, W. Jin, C. Wang, Y. Liu, S. Ma and X. Ren, “Jointly learning explainable rules for recommendation with knowledge graph,” in Proceedings of the World Wide Web Conference, pp. 1201–1221 (2019)
13) J. Bobadilla, F. Ortega, A. Hernando and J. Bernal, “A collaborative filtering approach to mitigate the new user cold start problem,” Knowledge-based Systems, vol. 26, pp. 225–238 (2012)
14) A. I. Schcin, A. Popescul, L. H. Ungar and D. M. Pennock, “Methods and metrics for cold-start recommendations,” in Proceedings of the 25th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 253–260 (2002)
15) M. Tiemann and S. Paew, “Towards ensemble learning for hybrid music recommendation,” in Proceedings of the 1st ACM Conference on Recommender Systems, pp. 177–178 (2007)
16) A. B. Barragán-Martínez, E. Costa-Montenegro, J. C. Burguillo, M. Rey-López, Marta, F. A. Mílec-Fonte and A. Peleteiro, “A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition,” Information Sciences, vol. 180, no. 22, pp. 4290–4311 (2010)
17) M. Kaminskas, F. Ricci and M. Schdi, “Location-aware music recommendation using auto-tagging and hybrid matching,” in Proceedings of the 7th ACM Conference on Recommender Systems, pp. 253–260 (2013)
18) X. Wang and Y. Wang, “Improving content-based and hybrid music recommendation using deep learning,” in Proceedings of the 22nd ACM International Conference on Multimedia, pp. 627–636 (2014)
19) D. Liang, M. Zhan and D. P. W. Ellis, “Content-aware collaborative music recommendation using pre-trained neural networks,” in Proceedings of the 16th International Society for Music Information Retrieval Conference, pp. 295–301 (2015)
20) R. S. Sutton and A. G. Barto, “Reinforcement learning: an introduction,” (2018)
21) L. P. Kaibbling, M. L. Littman and A. W. Moore, “Reinforcement learning: A survey,” Journal of Artificial Intelligence Research, vol. 4, pp. 237–285 (1996)
22) D. Sánchez-Moreno, A. B. G. González, M. D. M. Vicente, V. F. L. Batista and M. N. M. García, “A collaborative filtering method for music recommendation using playing coefficients for artists and users,” Expert Systems with Applications, vol. 66, pp. 234–244 (2016)
23) J. Chen, H. Zhang, X. He, L. Nie, W. Liu and T. Chua, “Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention,” in Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 335–344 (2017)
24) P. Darshana, “Music recommendation based on content and collaborative approach & reducing cold start problem,” in Proceedings of the 2nd International Conference on Inventive Systems and Control, pp. 1033–1037 (2018)
25) A. Ferraro, “Music cold-start and long-tail recommendation: bias in deep representations,” in Proceedings of the 2nd International Conference on Inventive Systems and Control, pp. 588–590 (2019)
26) K. Sakurai, R. Togo, T. Ogawa and M. Haseyama, “Music playlist generation based on reinforcement learning using acoustic feature map,” in Proceedings of the 9th Global Conference on Con-
Fig. 4 Recommendation performance comparison of PM and CMs 1-5.

S. Ji, S. Pan, E. Cambria, P. Marttinen and S. Y. Philip, “A survey on knowledge graphs: Representation, acquisition, and applications,” arXiv preprint arXiv:2002.00388, (2021)

Y. Zhang and X. Chen, “Explainable recommendation: A survey and new perspectives,” arXiv preprint arXiv:1804.11192, (2018)

F. Zhang, N. J. Yuan, D. Lian, X. Xie and W. Ma, “Collaborative knowledge base embedding for recommender systems,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 353–362 (2016)

X. Wang, D. Wang, C. Xu, X. He, Y. Cao and T. Chua, “Explainable reasoning over knowledge graphs for recommendation,” in Proceedings of the 33rd AAAI Conference on Artificial Intelligence, pp. 5329–5336, (2019)

X. Wang, X. He, Y. Cao, M. Liu and T. Chua, “KGAT: Knowledge graph attention network for recommendation,” in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 950–958 (2019)

X. Wang, T. Huang, D. Wang, Y. Yuan, Z. Liu, X. He and T.-S. Chua, “Learning intents behind interactions with knowledge graph for recommendation,” in Proceedings of the World Wide Web Conference, pp. 878–887 (2021)

S. Zhou, X. Dai, H. Chen, W. Zhang, K. Ren, R. Tang, X. He and Y. Yu, “Interactive recommender system via knowledge graph-enhanced reinforcement learning,” in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 179–188 (2020)

Y. Xian, Z. Fu, S. Muthukrishnan, G. De Melo and Y. Zhang, “Reinforcement knowledge graph reasoning for explainable recommendation,” in Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 285–294 (2019)

W. Song, Z. Duan, Z. Yang, H. Zhu, M. Zhang and J. Tang, “Explainable knowledge graph-based recommendation via deep reinforcement learning,” arXiv preprint arXiv:1906.09506, (2019)

A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” Advances in Neural Information Processing Systems, vol. 26, pp. 2787–2795 (2013)

S. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” The Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929–1958 (2014)

S. K. Nair, L. S. Thakur and K. Wen, “Near optimal solutions for product line design and selection: Beam search heuristics,” Management Science, vol. 41, no. 5, pp. 767–785 (1995)

G. Vigliensoni and I. Fujinaga, “The music listening histories dataset,” in Proceedings of the 18th International Society for Music Information Retrieval Conference, pp. 96–102 (2017)

M. Bakhshizadeh, A. Moeini, M. Latifi and M. T. Mahmoudi, “Automated mood based music playlist generation by clustering the audio features,” in Proceedings of the 9th International Conference on Computer and Knowledge Engineering, pp. 239–248 (2020)

M. Pyun, D. Kim, C. Lim, E. Lee, J. Kwon and S. Lee, “Examining the relationship between songs and psychological characteristics,” in Proceedings of the 22nd International Conference on Human-Computer Interaction, pp. 105–115 (2020)

L. Tong, “Personal recommendation based on community partition of bipartite network,” in Proceedings of the 2nd International Conference on Cloud Computing and Big Data, pp. 336–341 (2015)

Q. Ai, V. Azizi, X. Chen and Y. Zhang, “Learning heterogeneous knowledge base embeddings for explainable recommendation,” Algorithms, vol. 11, no. 9, pp. 137 (2018)

S. Rendle, C. Freudenthaler, Z. Gantner and L. Schmidt-Thieme, “BPR: Bayesian personalized ranking from implicit feedback,” arXiv preprint arXiv:1205.2618, (2012)

K. Zhao, X. Wang, Y. Zhang, L. Zhao, Z. Liu, C. Xing and X. Xie, “Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs,” in Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 239–248 (2020)
Fig. 5 Examples of the detailed recommendation results of the proposed method. In Example (1), PM recommends “Matter of Time” since it has similar acoustic features to “Move On” the target user has listened to. In Example (2), PM recommends “I Think I Smell A Rat” since it is created by “The White Stripes” that created “Seven Nation Army” the target user has listened to. In Example (3), PM recommends “Reckoner” since it has been listened to by another user who has listened to “Flume” that the target user has also listened to.

Keigo Sakurai received the B.S. degree in Electronics and Information Engineering from Hokkaido University, Japan in 2021. He is currently pursuing a M.S. degree with the Graduate School of Information Science and Technology, Hokkaido University. His research interests lie in the areas of machine learning application for data mining and information recommendation. He is a graduate student member of IEEE and a student member of IEICE.

Ren Togo received the B.S. degree in health sciences from Hokkaido University, Japan, in 2015, and the M.S. and Ph.D. degrees from the Graduate School of Information Science and Technology, Hokkaido University, in 2017 and 2019, respectively. He is also a Radiological Technologist. He is currently a Specially Appointed Assistant Professor with the Education and Research Center for Mathematical and Data Science, Hokkaido University. His research interests include machine learning and its applications. He is a member of IEEE.

Takahiro Ogawa received his B.S., M.S. and Ph.D. degrees in Electronics and Information Engineering from Hokkaido University, Japan in 2003, 2005 and 2007, respectively. He joined Graduate School of Information Science and Technology, Hokkaido University in 2008. He is currently an associate professor in the Faculty of Information Science and Technology, Hokkaido University. His research interests are AI, IoT and big data analysis for multimedia signal processing and its applications. He was a special session chair of IEEE ISCE2009, a Doctoral Symposium Chair of ACM ICMR2018, an organized session chair of IEEE GCC2017-2020, a TPC Vice Chair of IEEE GCC2018, a Conference Chair of IEEE GCC2019, etc. He has been also an Associate Editor of ITE Transactions on Media Technology and Applications. He is a senior member of IEEE and a member of ACM, IEICE and ITE.

Miki Haseyama received her B.S., M.S. and Ph.D. degrees in Electronics from Hokkaido University, Japan in 1986, 1988 and 1993, respectively. She joined the Graduate School of Information Science and Technology, Hokkaido University as an associate professor in 1994. She was a visiting associate professor of Washington University, USA from 1995 to 1996. She is currently a professor in the Faculty of Information Science and Technology, Hokkaido University. Her research interests include image and video processing and its development into semantic analysis. She has been a Vice-President of the Institute of Image Information and Television Engineers, Japan (ITE), an Editor-in-Chief of ITE Transactions on Media Technology and Applications, a Director, International Coordination and Publicity of The Institute of Electronics, Information and Communication Engineers (IEICE). She is a fellow of ITE, a senior member of IEEE and a member of IEICE and ASJ.