Technology Reports (Special Article) Recommendation Deep Learning Music
Special Articles on Lifestyle-enriching AI Technology

Music Recommendation Technology that Considers the Time Series of Singing History

Service Innovation Department Shigeki Tanaka Yusuke Fukazawa†

While karaoke is replete with search functions that users can use to select music (by singer, by era, etc.), it is rare for a system to have functions to actively make song recommendations. This is because making recommendations in karaoke is particularly difficult because the subject of the recommendations is a group, there is no singing history available from past visits, and the songs selected change due to the atmosphere. To address these issues, NTT DOCOMO has developed a deep learning recommendation model. This deep learning model considers a time series and learns in conjunction with various music information to make suitable recommendations to an unknown group.

1. Introduction

With the advances in machine learning[*1], recommendation technologies have been introduced to various services such as online shopping, that have contributed to improved user experiences and business invigoration. In contrast, making suitable recommendations with karaoke is extremely difficult because karaoke users have the following tendencies:

- Users are a group with no singing history
- Group members select songs based on the composition of the group or the atmosphere in the venue

The collaborative filter recommendation method is a typical method of recommendation and used in various areas. In this method, the interests and preferences of an individual is inferred from the user's histories such as product purchase history or Web browsing history, and products that similar users are interested in are recommended to

† Currently, General Affairs Department

the user as related products. A method of combining individual profiles to expand them to a group profile has also been proposed [1]. Meanwhile in karaoke, many users have no singing history from past visits because they didn't log into the karaoke terminal, so not only is profiling difficult, even the number of people in the group is unknown.

For this reason, as an example, Daiichi Kosho provides a recommendation function that recommends music similar to the last tune that was sung. However, when using karaoke in a group, group members take turns to sing, so even though it is possible to recommend music to the member who sang most recently, if the subsequent member is different, effective recommendation is not possible.

Recommendation is also difficult because members select songs to suit the members of the group or the atmosphere. For example, with groups consisting of coworkers, or friends or families, there is a high probability that the songs likely to be selected will be different. Also, the atmosphere in a karaoke studio changes from the time people enter to when they leave, and many people like to sing a particular song when the atmosphere becomes exciting, or a closing song at the end. Thus, in many cases, the atmosphere of the venue affects song selection, so it is important to be able to read such changes when making recommendations. However, recommendation methods typical of the aforementioned collaborative filters generally capture tastes and preferences over a long term, and therefore do not hold much promise for effectiveness in areas that are constantly changing due to the venue or the atmosphere [2].

To address these issues, NTT DOCOMO has developed a recommendation model that uses deep learning*2.

This model predicts the most likely song or singer that will be subsequently selected by considering a time series and based on various song metadata when a song is selected. This makes it possible to make effective recommendations to groups who have no past history, and also makes it possible to make recommendations that consider the atmosphere of the venue by capturing the flow of songs.

This article describes details of the proposed method, and describes real-time song recommendation engine using this method.

2. Deep Learning Recommendation Model Considers Time Series and Multivariates

NTT DOCOMO's proposed recommendation model recommends subsequent songs or singers through input of various metadata of the song previously reserved.

2.1 Model Overview

Figure 1 shows an overview of the proposed model.

In addition to the ID of the song previously sung, the model inputs information such as the singer name, the composer name and the music genre and converts these into feature values. For example, the singer feature value represents the level of similarity and so forth between singers and the model acquires the feature values that better represent singers through learning. These feature values are input to a Recurrent Neural Network (RNN). RNN has a function that holds information called a hidden state, performs output computation

*1 Machine learning: A framework that enables a computer to learn the relationships between inputs and outputs by statistical processing of examples.

*2 Deep learning: Machine learning using a neural network with a multi-layered structure.

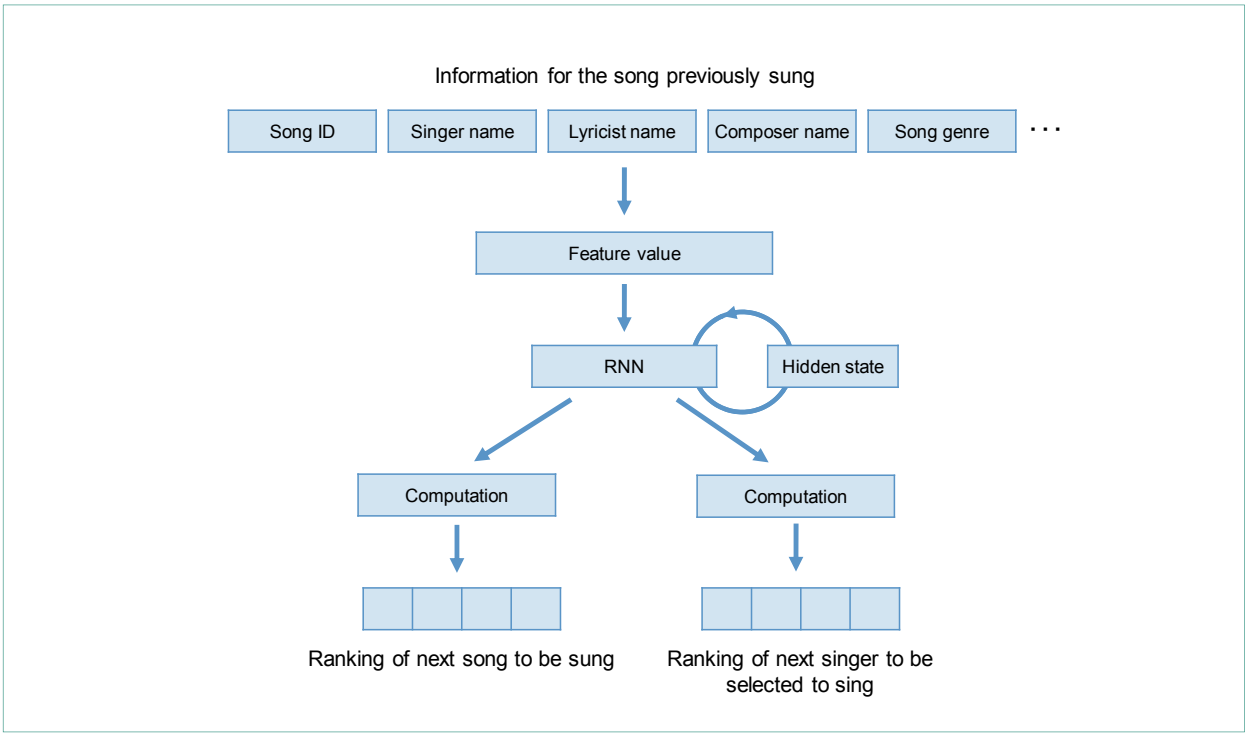

Figure 1 Overview of the proposed recommendation model

from the input feature value and the hidden state, and at the same time updates the hidden state. This mechanism makes it possible to take into account information that was input in the past. Using the RNN output, the model generates rankings for subsequent songs and singers.

The features of this model are that it (1) uses RNN, (2) inputs diverse song information, and (3) simultaneously performs multiple tasks (outputs rankings for subsequent songs and singers).

(1) RNN makes it possible to store a certain amount of information input so far. This makes it possible to capture not only the most recently entered song information, but also capture a time series history from the first song reserved after the group entered the karaoke studio to the song reserved right before recommendation, to enable recommendation that takes into account whose turn it is to sing, etc.

(2) Generally with recommendations, only song IDs are handled as input. In contrast, because understanding of songs is deeper in this model due to its handling of a wider range of song information, the model potentially has better recommendation accuracy, which is the probability that a song a member is about to sing is included in the recommendation results.

(3) Multitasking enables the improvement of resource utilization efficiency. Generally, machine learning models learn specifically for a single task, and if there are multiple tasks, more individual models are generated. Thus,

resource consumption for model learning and model operations increases in proportion to the number of tasks. Design to solve multiple tasks simultaneously is known as "multitask learning," and entails the model learning over a wider field to solve the related multiple tasks it has been given, which contributes to the improvement of accuracy for each task. Furthermore, by implementing two functions in one model, we were able to halve resource consumption compared to the cases where models for each function are prepared.

2.2 Verifying Accuracy

To verify the accuracy of our proposed method, with the cooperation of Daiichi Kosho, we compared it with existing methods using one month of data from a karaoke business. The first three weeks of data were used for each model to learn, then in the last week, we compared the estimation accuracy of each model.

For the existing methods, the following two methods were used.

- Ranking: Always recommends songs that are popular in the learning period.
- Item-kNN (k-Nearest Neighbor Algorithm): An item-based collaborative filter recommendation method that pre-creates a table of similarities between songs during the learning period. This table shows how likely it is for the same user group to sing those respective songs, and the method recommends songs similar to songs sung most recently.

The accuracy comparison with MAP@N[*3] resulted in **Figure 2**, and showed that the accuracy of the proposed method exceeding the existing methods by more than 10% in both song and singer recommendations.

Figure 3 shows the movement of accuracy through time by dividing the history from customer entry through to exit of the karaoke studio into three parts and evaluating the averages of

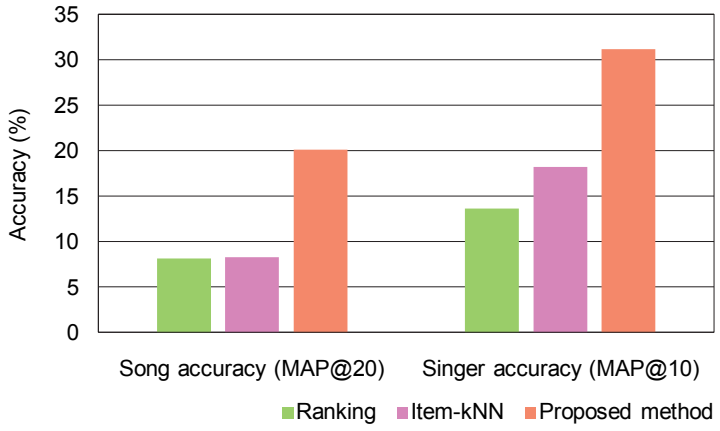

Figure 2 Comparison of accuracy of models

*3 MAP@N: When recommending N items, the probability that the next item selected by the user is included in those items.

accuracies of those respective periods. From the decrease over time of ranking accuracy after entering the room, it can be seen that in general, the user group sings a popular song at the beginning, and then gradually select a variety of songs. Also, Item-kNN accuracy was extremely low after entering the room, rose in the middle and then fell again, which presumably shows tendencies for individual group members to select songs they like at the beginning, to be affected by other members' song selections in the middle, and then finally members to sing their closing song. Compared to both these models, it can be seen that our proposed method achieves greater accuracy in all three periods, and captures changes in the atmosphere of the group.

3. Real-time Song Recommendation Engine

The real-time song recommendation engine was built on Amazon Web Services (AWS)*4 as the system using the proposed model. We are providing these recommendation functions to Daiichi Kosho via an Application Programming Interface (API)*5.

Figure 4 shows an overview of the system.

This system is constructed with a batch server and an API server. The batch server periodically relearns recommendation models from singing history and song information stored in a database, and then sends the learned models to the API server which updates the learned models with the latest ones. Although the batch server is not running except during the above periodic execution, the API server is always running to process recommendation requests. When the API server receives a recommendation request, it performs computations with the learned model based on terminal information and singing history sent as parameters, and then responds with the subsequent song and singer recommendations. While the database singing history is not necessarily updated in real time, real-time update of recommendations on reserving

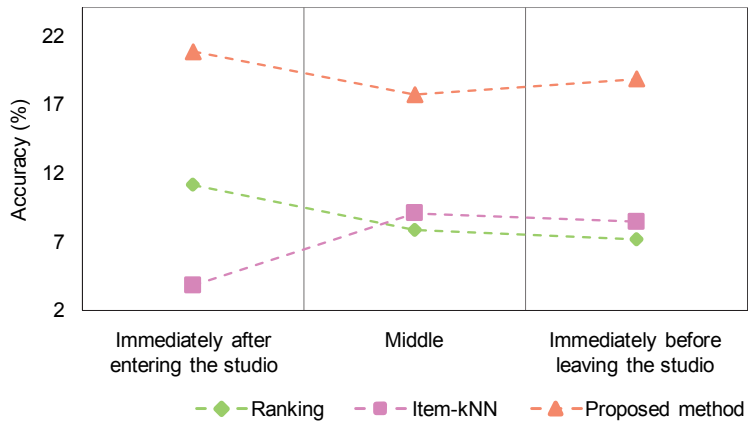

Figure 3 Movements in song accuracy over time for each model (MAP@20)

*4 AWS: A cloud computing service provided by Amazon.com.

*5 API: An interface that enables software functions to be used by another program.

songs is possible because the API server accepts the singing history on the terminal with each recommendation request.

As shown in **Figure 5**, the recommendation function is provided on the Daiichikosho Amusement Multimedia (DAM) terminal pre-reservation screen. Recommendations based on the history so far are shown by tapping on the relevant tab.

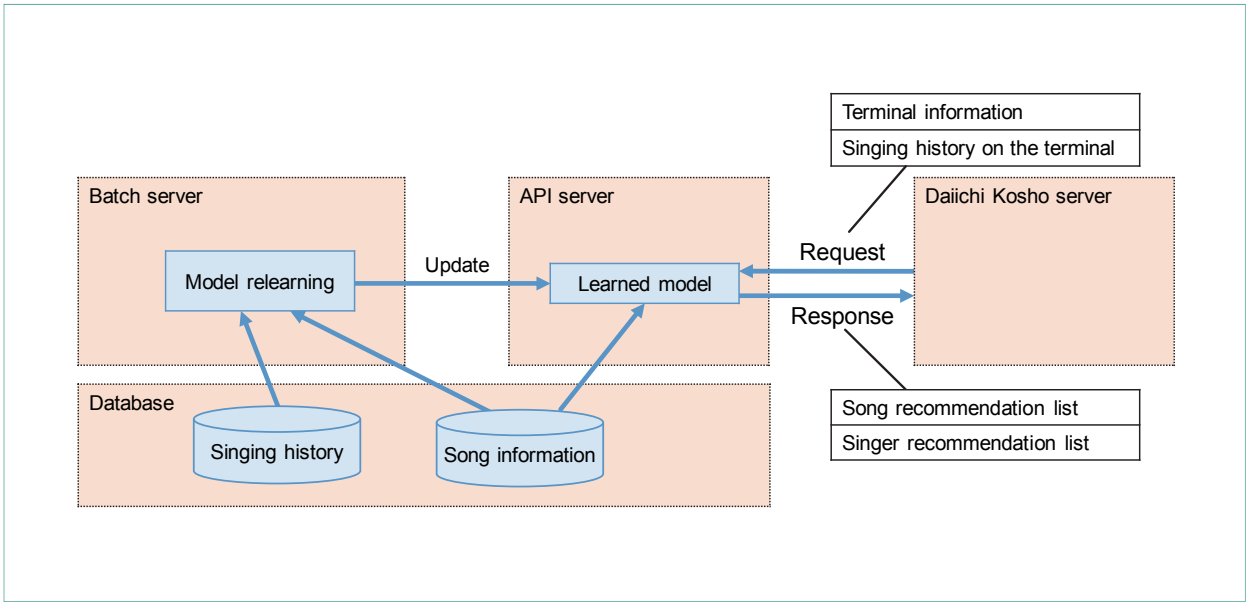

Figure 4　Overview of the recommendation system

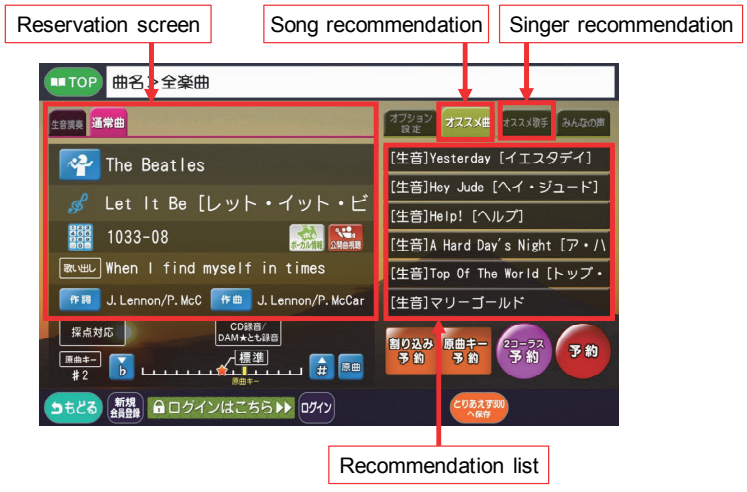

Figure 5　Recommendation display screen on the karaoke terminal

4. Conclusion

This article has described recommendation issues with karaoke systems, a recommendation method that uses deep learning to solve those issues, an evaluation of the performance of the method, and the recommendation system that uses the method.

At the time of writing this article, recommendation functions were being provided to a karaoke business as a trial, and improvements were being made while receiving feedback. In the future, we hope to contribute to user experiences with a formal commercial service widely in use.

REFERENCES

[1] A. Felfernig, L. Boratto, M. Stettinger and M. Tkalčič: "Group recommender systems: An introduction," Springer, 2018.

[2] M. Pazzani and D. Billsus: "Learning and revising user profiles: The identification of interesting web sites," Machine learning, Vol.27, No.3, pp.313-331, 1997.