User Modeling and User-Adapted Interaction (2022) 32:355–388
https://doi.org/10.1007/s11257-022-09325-y

An automated system recommending background music to listen to while working

Hiromu Yakura[1,2] · **Tomoyasu Nakano**[2] · **Masataka Goto**[2]

Received: 4 January 2021 / Accepted in revised form: 6 March 2022 / Published online: 18 May 2022

Abstract

Many people listen to music while working nowadays. However, conventional recommendation systems that are designed for playing songs matching user preferences cannot be applied for such a situation. This is because previous research showed that listeners' concentration can be negatively affected not only by music that listeners strongly dislike but also by music that the listeners strongly like. Therefore, when we consider a recommendation system to be used while working, it is desirable to avoid both songs the user likes very much and songs the user dislikes very much. Given this background, we propose *FocusMusicRecommender*, a system designed specifically for recommending music to listen to while working. It summarizes songs automatically and plays them successively in order to enable users to give not only "dislike (very much)" feedback via a "skip" button but also "like (very much)" feedback via a "keep listening" button. The feedback is then combined with the users' concentration level that is estimated from their behavioral history during the playback of the corresponding song, which allows the system to obtain preference information that distinguishes between "like" and "like very much" without burdening the user who is working. Based on the preference information, the system estimates the preference levels of unplayed songs and prioritizes the songs for subsequent playback by also considering the user's current concentration level. Our experiments showed the validity and effectiveness of the proposed method, including the accuracy of the concentration level estimation. Moreover, our user study verified the suitability of the recommendation results from both the observed behavior and obtained comments of the participants.

This work was supported in part by JST ACCEL Grant Number JPMJAC1602, JST ACT-X Grant Number JPMJAX200R, JST CREST Grant Number JPMJCR20D4, and JSPS KAKENHI Grant Number JP21J20353.

✉ Hiromu Yakura
hiromu.yakura@aist.go.jp

[1] University of Tsukuba, Tsukuba, Japan

[2] National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan

Keywords Music recommender · Background music recommendation · Concentration level estimation

1 Introduction

Recommendation is a fundamental technique to deliver a personalized media experience (Park et al. 2012; Lu et al. 2015; Zhang et al. 2019; Deldjoo et al. 2020). To entertain a user who consumes media content as their main task (sole activity), many methods have been proposed to find content matching the user's interests or preferences adaptively. Simultaneously, media content is often used as a means to achieve purposes other than the consumption of the content itself. For such a second task situation, conventional recommendation systems for pursuing user satisfaction with the content itself are not always suitable.

Specifically, we consider a situation where people are listening to music as their second task to concentrate on their main task, such as working or studying. In fact, this is a common practice nowadays; Lonsdale and North (2011) surveyed 189 university students and reported that 75.7% of them confirmed that they had used background music while working or studying. Additionally, most of those respondents mentioned the effectiveness of music in helping them concentrate, making comments, such as "background music stops my mind from wandering when I need to focus on work." We argue that conventional recommendation systems (Song et al. 2012; Knees and Schedl 2013) are not suitable for use during work.

This is because a song strongly preferred by a user may interfere with the user's concentration, as reported by Huang and Shih (2011). They investigated the relationship between concentration level while listening to songs in different genres and preference level for the songs, which was rated on a five-point Likert scale of "like very much," "like," "neither like nor dislike," "dislike," and "dislike very much." They found that the concentration level measured by attention testing dropped significantly not only when people were listening to songs they disliked very much but also when people were listening to songs they liked very much. In contrast, the concentration levels of people who listened to songs they liked, neither liked nor disliked, or disliked did not significantly differ from those of people in a silent environment.

This implies that there is a gap between the current music recommendation systems and a user's motivation to use them. The user wants the systems to select suitable songs automatically because selecting songs while working is troublesome. However, the systems are designed to find and play songs the user likes very much although avoiding songs that evoke strong emotions is important for concentrating on work. We therefore propose *FocusMusicRecommender*, a system specifically designed to recommend suitable background music for listening while working. It thus not only prioritizes songs a user may neither like nor dislike based on the user's preference information but also collects the preference information without interfering with the user, as depicted in Fig. 1.

More specifically, FocusMusicRecommender introduced various interaction techniques so that it works without asking the user in work to rate each song in a way similar to that required by the conventional systems. Firstly, it plays songs in an

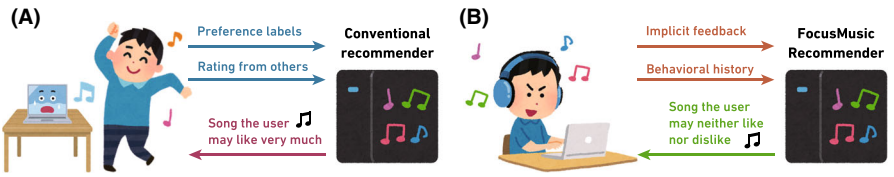

Fig. 1 (**A**) According to the study by Huang and Shih (2011), conventional recommendation systems trying to play a song that a user likes very much may distract the user. (**B**) FocusMusicRecommender helps a user concentrate on work by automatically selecting a song that the user would neither like nor dislike

abridged manner unless the user presses a "keep listening" button, which makes it possible to infer the user's positive preference, such as "it is my favorite song and I want to listen to it more," when they use the button. Analogously, it can infer the user's negative preference, such as "I dislike this song and want to skip it," when the user presses a "skip" button. This implicit feedback enables FocusMusicRecommender to determine the user's preference level of "like (very much)," "neither like nor dislike," or "dislike (very much)."

Still, it is not precise enough considering the report from Huang and Shih (2011), that is, a song the user likes very much and that the user likes have different impacts on their concentration. Hence, FocusMusicRecommender incorporates a mechanism to estimate the user's concentration level from the working behavior and uses it to distinguish the case when the user presses the "keep listening" button because the user likes a song very much from the case when the user moderately likes the song. This refinement process is based on the hypothesis that the feedback given when concentrating reflects the preference level for songs more faithfully than the feedback given when not concentrating. Furthermore, by applying machine learning for the collected preference information of each user, FocusMusicRecommender can select songs the user may neither like nor dislike without further feedback once the user listened to a certain number of songs.

To evaluate the effectiveness of FocusMusicRecommender, we first conducted three experiments regarding the accuracy of the concentration level estimation, the validity of the preference level determination from the implicit feedback, and the generality of the playback interaction. We also conducted a user study with four comparison implementations to confirm the suitability of the recommendation results and their effect on the users. Our results from the experiments and user study supported the effectiveness of FocusMusicRecommender, which illuminates a new way of supporting media consumption using recommendation systems.

This article is an extended version of our previous conference paper (Yakura et al. 2018). In addition to having a more detailed presentation of the design (Sect. 3) and the implementation (Sect. 4) of the proposed method, this article complements the previously published findings with new evaluation results. For instance, we expanded the analysis of the result of the concentration level estimation in Sect. 5.2 to understand the reasons behind its relatively better accuracy than previous methods. This analysis revealed that employing n-gram-based features and exploiting Web communication history, both of which were not examined in previous methods, were effective to improve the accuracy without using external sensors. As described in Sect. 5.4, we also

conducted an additional experiment to evaluate the usage of the proposed interaction technique, which confirmed its generality in determining the preference level without burdening users. Furthermore, we added a new comparison implementation in the user study (Sect. 6.1), which makes it possible to examine the effects of the played songs on the users, as presented in Sect. 6.3. Based on the above results, we newly discussed limitations of FocusMusicRecommender and its further directions in Sect. 7 to present possibilities of new music-driven interactions that consider the user's concentration level.

2 Related work

In this section, we first explain previous findings regarding the effect of background music on listeners' concentration level, which motivated us to introduce FocusMusicRecommender. We then introduce related studies on music recommendation and concentration level estimation. For music recommendation, we focus on systems designed for specific purposes or based on limited feedback considering the usage of the proposed system.

2.1 Effect of background music on listeners' concentration

As mentioned in Sect. 1, music has the power of not only entertaining people who listen to it as their main task but also affecting people who listen to it as their second task. For example, music is used with videos to make them more immersive or used in restaurants to bring out the flavors of food (Milliman 1986; Biswas et al. 2018). Its power to help listeners concentrate has also been investigated by many researchers (Mendes et al. 2021) from a pedagogical (Hallam et al. 2002) and management (Fox 1971) perspective.

In this context, Huang and Shih (2011) measured that the concentration level of participants scored by an attention test while listening to songs in different genres, such as classical and popular music. They analyzed the relationship of the scores to the participants' self-reported preference levels of the played songs. Compared with a silent environment, their result showed that the scores significantly dropped not only when people were listening to songs they disliked very much ($p < 0.05$) but also when listening to songs they liked very much ($p < 0.01$). Furthermore, there was no significant effect of the genres of the songs, such as whether the song is classical or popular music, on the concentration levels. As a result, they concluded that the listener's concentration level depends more on the listener's preference level than on the song's musical genre.

This point introduces a new perspective for the existing studies focusing on the relationship between background music and listeners' concentration level (Mendes et al. 2021). For instance, classical compositions of Mozart have been conventionally considered to help listeners concentrate, which is sometimes referred to as the "Mozart effect" (Ho et al. 2007), but it can be explained from the fact that most people have a moderate preference for the compositions and their emotions would not be deeply

aroused. This gives room for leveraging recommendation systems to help listeners concentrate. That is, rather than playing Mozart's compositions for all users, playing songs that each user may not like or dislike very much based on their preference information would be helpful.

Here, without such recommendation systems, it is possible that users naturally choose songs that would not evoke strong emotions under the context that they listen to the songs while working. This seems reasonable considering that, in Huang and Shih (2011), participants rated their preference levels to the played songs without the context, i.e., how much they like each song in general. Meanwhile, Johansson et al. (2011) implied the difficulty of choosing optimal songs to be listened to while working themselves. Specifically, Johansson et al. (2011) investigated participants' levels of emotional arousal while listening to songs they picked as those they want or do not want to listen to while working on their choice. Consequently, they found that the levels measured via the participants' pupil size were significantly elevated more than that of a silent environment regarding both the songs they wanted and did not want to listen to. Therefore, considering the conclusion of Huang and Shih (2011), we cannot rule out the possibility that songs chosen by users to listen to while working still interfere with them.

Generally, even if we assume that users know the impact of listening to songs they like very much while working, demanding them to judge whether each song is suitable to be listened to while working would not be desirable according to the dual process theory (Kahneman 2011). That is, this judgment requires a reasoning process based on the prior knowledge, involving "System 2" of their brain, and thus consumes their attention while working. This point led us to design FocusMusicRecommender to work with users' "System 1"-based interactions (i.e., implicit feedback based on their intuitive preference) in order not to interfere with them. We note that, since the term "attention" has ambiguity as it sometimes refers to the relative assignment of one's cognitive resource between multiple tasks (Proverbio et al. 2015) based on Kahneman's capacity model (Kahneman 1973) or one's ability to maintain a focus on a single main task, we hereinafter follow the terminology of Huang and Shih (2011) in using "concentration" in this article.

2.2 Music recommendation for specific purposes

There are some studies of music recommendation systems designed for specific purposes (Oliver and Flores-Mangas 2006; Liu et al. 2010; Baltrunas et al. 2011). For example, Oliver and Flores-Mangas (2006) proposed MPTrain, which is intended to facilitate physical exercise by playing music. These systems play faster songs when the heart rate is lower than the desired workout, and vice versa. Baltrunas et al. (2011) proposed InCarMusic, which is designed to assist car drivers by changing music genres to be played according to whether the user is sleepy or not, whether the user is traveling on ordinary road or in expressway, and so on.

However, to the best of our knowledge, there is no study on music recommendation systems specially designed for use while working, even though many new recommendation methods using recent machine learning algorithms have been proposed (van den

Oord et al. 2013; Liang et al. 2015; Wang 2020). We acknowledge that Volokhin and Agichtein (2018) proposed a method for automatically generating activity-specific playlists, including ones for working, but there is no special consideration or evaluation regarding helping users concentrate.

We note that these recommendation systems designed for specific purposes, including FocusMusicRecommender, can be aligned with context-aware recommendation systems (Adomavicius and Tuzhilin 2015), as they leverage additional contextual information to recommend optimal songs reflecting their purposes. In this sense, we can apply existing methods for context-aware recommendation systems to recommend songs that a user may "feel right" to listen to while working by collecting information about the user's working context, in a manner similar to Kaminskas and Ricci (2017) did to recommend songs that a user may "feel right" to listen to when they visit a specific place by evaluating candidate songs with location-compatible emotion labels. However, as discussed in Sect. 2.1, we must carefully consider how to collect such contextual information from users in working since demanding them to judge each song whether they "feel right" to listen to it while working would interfere with them. This point motivated us to design a recommendation system dedicated to finding songs to be listened to while working, rather than applying existing techniques for context-aware recommendation systems.

2.3 Music recommendation with limited feedback

As we mentioned in Sect. 1, considering the use while working, asking users to give explicit feedback for each song is unrealistic because it would interfere with their main tasks. In this respect, Pampalk et al. (2005b) proposed a music recommendation system that assumes limited implicit feedback. It uses a metric calculated based on skip operation, i.e., whether the user skipped or not while a song was playing, as follows:

1. For each candidate song, let s_s be the musical similarity to the nearest song skipped so far, and let s_a be the similarity to the nearest song that was not skipped.
2. If there are candidate songs satisfying $s_a > s_s$, select from them the one having the largest s_a.
3. Otherwise, select from all the candidate songs the one with the largest $\frac{s_a}{s_s}$.

Here, the musical similarity is calculated from audio signals based on spectral features and fluctuation patterns (Pampalk et al. 2005a).

Pampalk et al. (2005b) described that this metric was designed to reflect two levels of preference information, which is informed by regarding skipped songs as disliked and songs that were not skipped as liked. Thus, it cannot be used for recommending songs to listen to while working because it requires giving priority to songs that the user may neither like nor dislike.

2.4 Automatic estimation of concentration level

Helping users to concentrate while they are using computers is one of the topics actively discussed in human–computer interaction. Specifically, while users are often interrupted by emails or prompts from applications (Czerwinski et al. 2004), such interruptions evoke their stress and frustration (Mark et al. 2008). In this regard, several studies (Tateyama et al. 2004; Fogarty et al. 2005; Züger and Fritz 2015; Tanaka and Fujita 2011) have tried estimating the "concentration" or "interruptibility" level of users working on a personal computer, which can be leveraged to help users concentrate, as FocusMusicRecommender does.

For example, Tateyama et al. (2004) proposed a method for estimating based on eye movements tracked by a stereo camera. Fogarty et al. (2005) employed the numbers of mouse and keyboard operations as well as the number of door openings that are counted using a magnetic sensor. Züger and Fritz (2015) exploited physiological responses, such as the user's skin potential or heart rate, and concluded that these metrics are useful for estimating the interruptibility of workers. Tanaka and Fujita (2011) presented a method that incorporates not only the number of mouse and keyboard operations but also the number of switching of the active application.

However, considering costs and psychological barriers, the use of external sensors is unrealistic for music recommendation. Additionally, though Tanaka and Fujita (2011) proposed a method that does not use external sensors, it employs the number of specific operations within a fixed length of time and therefore is hard to use when the playback duration is variable, as a song can be skipped after only a few seconds.

3 FocusMusicRecommender

In this section, we present an overview of FocusMusicRecommender (Fig. 2) and describe the details of its three core components that reflect our design strategy for the system to be used while working: how to determine the preference levels of played songs (Sect. 3.2), how to select the next song to be played (Sect. 3.3), and how to automatically estimate the concentration level (Sect. 3.4).

3.1 Overview

FocusMusicRecommender aims to help users concentrate when listening to music while working on a personal computer. In this case, an automatic playback function is often used to save time and effort in selecting songs during work. However, as mentioned in Sect. 1, the random playback and conventional recommendation methods would play songs the users like very much, which would interfere with their concentration. Instead, this system automatically selects songs the user may neither like nor dislike very much and plays them consecutively.

To deliver such songs without asking the user in work to give explicit feedback, the proposed system introduces a summarized playback using chorus section information. In this system, the song is terminated after its first chorus section unless the user

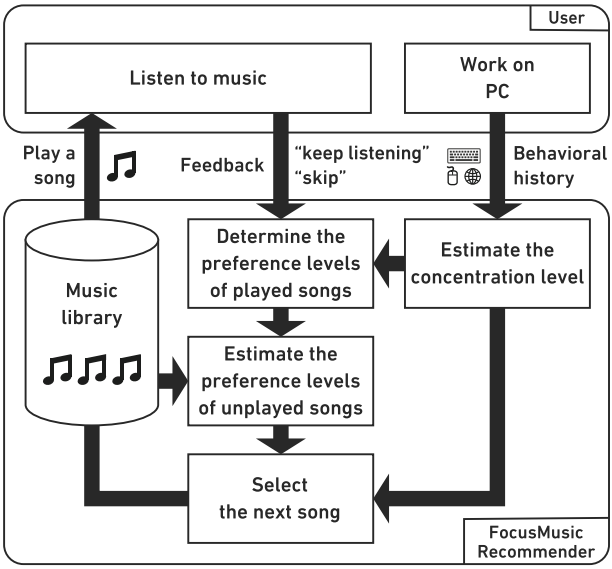

Fig. 2 Overview of FocusMusicRecommender. It determines the user's preference levels of songs and selects the song played next according to the user's feedback and behavioral history

presses a "keep listening" button. It allows the system to determine the user's positive preference for the played song, in the same framework as proposed by Pampalk et al. (2005b) of acquiring the negative preference by a "skip" button. In other words, without this summarized playback, it is difficult for the proposed system to avoid playing songs the user may like very much because it cannot know which song is preferred by them.

Then, the proposed system looks for songs that would be neither liked nor disliked from unplayed songs based on the collected preference information for the played songs. However, the preference information in three levels obtained from pressing either of the two buttons or neither of them would not be precise enough to our aim of avoiding songs the user likes very much or dislikes very much. The system therefore estimates the current level of the user's concentration and uses it to refine the preference information by considering the user's situation behind the implicit feedback. The estimated concentration level is also exploited to adjust the selection criterion of the next song from candidate songs judged as to be neither liked nor disliked for the purpose of helping the user to concentrate more.

3.2 Determine the preference levels of played songs

To begin with, we explain in detail how the proposed system determines the user's preference level for played songs. As shown in Fig. 3(B), we first present the "keep listening" button with the summarized playback to determine in the three levels of "like (very much)," "neither like nor dislike," and "dislike (very much)." We then extend the determination to five levels by leveraging the user's concentration level, as depicted in Fig. 3(C).

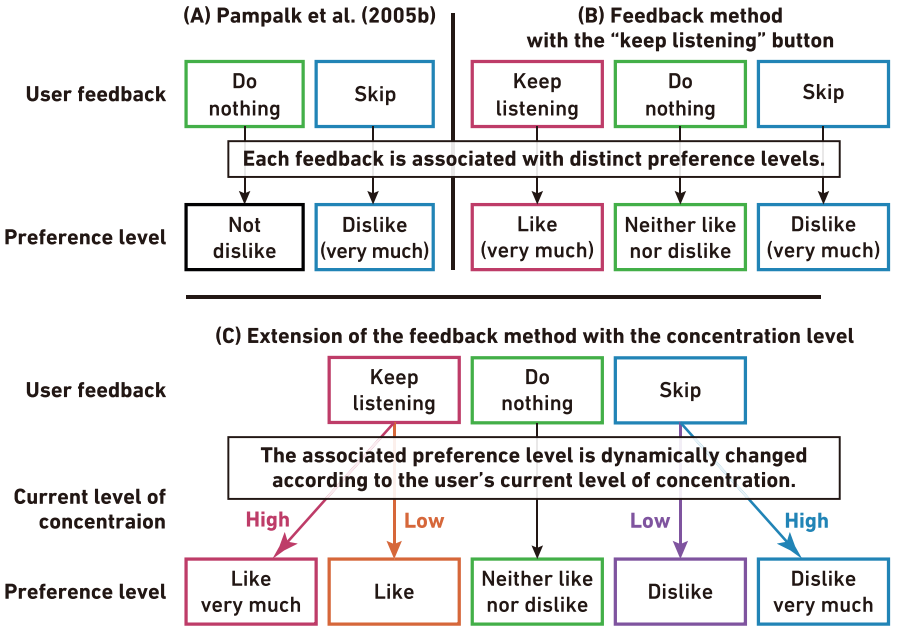

Fig. 3 Relationships between user feedback and determined preference levels. (**A**) From the "skip" feedback, only "dislike (very much)" or not can be distinguished. (**B**) The addition of "keep listening" feedback gives three preference levels, including "like (very much)." (**C**) Combining the concentration level gives five levels distinguishing between "like very much" and "like" as well as between "dislike very much" and "dislike"

3.2.1 Obtain feedback using a "keep listening" button

As described in Sect. 2.3, the approach of Pampalk et al. (2005b) can determine the user's preference level from the implicit feedback. However, the level is binary: "dislike (very much)" or not, as it is informed by skipped or not. The proposed system therefore extends this approach by introducing the "keep listening" button with a summarized playback, which enables the system to distinguish songs that the user likes or likes very much from the other songs (i.e., songs the user neither likes nor dislikes and songs the user dislikes or dislikes very much). For the automated summarization, it uses the chorus section information of the played song, inspired by previous literature (Logan and Chu 2000; Cooper and Foote 2002; Dannenberg and Goto 2008) and its use in portable music players[1]. Then, in accordance with the user feedback, we determine the preference levels as follows:

- **Press the "skip" button**
  When the user presses the "skip" button, the system presumes that the user dislikes the song, potentially very much, and immediately switches to the next song.
- **Press the "keep listening" button**
  When the user presses the "keep listening" button, the system presumes that the user likes the song, potentially very much, and plays it to the end.

[1] ZAPPIN Playback: https://docs.sony.com/release/NWZW273S_W274S_guide_EN.pdf.

– **Do nothing**
When the user does not press any buttons until the end of the first chorus section, the system presumes that the user neither likes nor dislikes the song and plays the next song. We note that, since it is difficult for the user to decide whether to skip or keep listening to the song right after it starts playing, we designed the system to play each song for at least 30 seconds.

3.2.2 Refine the preference level with estimated concentration level

The preference levels obtained in "like (very much)," "neither like nor dislike," and "dislike (very much)" are subsequently refined into five levels distinguishing "like very much" from "like" and "dislike very much" from "dislike" based on the user's estimated concentration level under the following hypothesis.

> **Hypothesis** User feedback obtained under a high level of concentration expresses the preference level better than feedback obtained under a low level of concentration.

In other words, a concentrating user would not press the "keep listening" or "skip" button unless the user likes or dislikes the song strongly. Thus, as presented in Fig. 3 (C), the preference level is refined in accordance with each combination of the user feedback and concentration level.

The motivation for this refinement process is that distinguishing songs liked and those liked very much is important for our purpose. This is because avoiding songs the user may like very much is particularly crucial since they can interfere with the user's concentration. In other words, according to the results of Huang and Shih (2011), misclassifying songs liked as songs neither liked nor disliked may not have a significant impact on the user's concentration since both preference levels would not distract the user. Analogously, misclassifying songs disliked as songs neither liked nor disliked may also not have a significant impact. However, if the system misclassified songs liked very much as songs neither liked nor disliked or songs disliked very much as songs neither liked nor disliked, it can lead to the user's distraction by playing songs liked very much or disliked very much instead of songs liked or disliked, respectively.

This refinement process is further beneficial when we employ hierarchical classification algorithms (Silla Jr. and Freitas 2011) instead of regular classification or regression models. More specifically, it allows us to explicitly incorporate such prioritization among the five-level preference, as shown in Fig. 4. Here, the algorithms put a greater penalty on misclassifying songs liked very much as songs liked than on misclassifying songs neither liked nor disliked as songs liked. This would reduce the risk of misclassifying songs that the user likes or dislikes very much as songs suitable to be listened to while working.

3.3 Select the next song considering the concentration level

Through the above processes, the proposed system determines preference levels of played songs and accumulates them. Relative to the collected information, the system

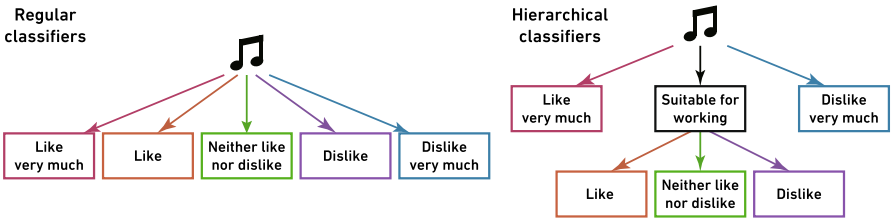

Fig. 4 Difference between regular and hierarchical classifiers in FocusMusicRecommender. By extending the feedback into five levels, we can introduce the hierarchical classifiers to regard liked songs, neither liked nor disliked songs, and disliked songs as songs suitable to be listened to while working

estimates the preference levels of songs that have not been played yet based on a musical similarity among the songs. Specifically, it applies a machine learning algorithm to a set of pairs of the musical features, which were extracted from each played song, and its corresponding preference level. It then can estimate the preference level of each unplayed song from its musical features and play songs estimated to be ones that the user may neither like nor dislike very much. If there are two or more such songs, the system has to choose one of them.

We designed the system so that in such a situation it would adjust the selection criterion according to the current level of concentration in a way that would help the user concentrate more. When the concentration level is high, the system prioritizes songs similar to the song played immediately before to avoid making sudden changes that can distract the user. Conversely, when it is low, the system tries giving the user a chance to change their mood by playing various songs. This is based on the survey by Wells (1990): While 73.3% of 225 respondents confirmed that music helps them change their mood, the genres of music they listen to for the mood change varied widely.

3.4 Estimate the concentration level

To achieve the preference level refinement (Sect. 3.2.2) and song selection (Sect. 3.3), the system must determine whether the user is concentrating. Therefore, the system estimates the concentration level automatically by applying a machine learning algorithm to the user's behavioral history during the last song (see Sect. 4.2) in the same manner as previous methods described in Sect. 2.4. As we will describe later in Sect. 4.2, we used AROW (Crammer et al. 2013) for the machine learning algorithm, though other algorithms can also be used as long as our purpose is achieved.

The system collects three types of behavioral history: keyboard input, mouse input, and Web communication (Table 1). Since, as mentioned in Sect. 2.4, using external sensors for music recommendation is unrealistic, the system uses software-based features, as shown in Table 2. Here, we introduce Web communication history, which has not been used in previous methods (Fogarty et al. 2005; Züger and Fritz 2015; Tanaka and Fujita 2011) because Web communication reflects the work content, such as whether the user is searching on the Web or using social networking services. This feature would be useful for the concentration level estimation. For example, the user accessing social network services often might be distracted.

Table 1 User behaviors used by FocusMusicRecommender for the concentration level estimation

Types of Behavior	Format	Examples
Keyboard input	"key *Target application name* *Key*" (The use of modifier keys is merged into a single event.)	"key Microsoft Excel a" (Press "a" in Microsoft Excel) "key Mozilla Firefox <[Ctrl: v]>" (Press "Ctrl+v" in Mozilla Firefox)
Mouse input	"mouse *Target application name* *Event number*" (The event number distinguishes between three types of clicks and scrolling in four directions.)	"mouse Google Chrome 1" (Click a left button in Google Chrome) "mouse Eclipse 4" (Scroll up in Eclipse)
Web communication (HTTP/HTTPS)	"web *Request method* *Hostname*" (Only GET and POST requests are collected.)	"web GET www.google.com" (Send a GET request to www.google.com) "web POST twitter.com" (Send a POST request to twitter.com)

Table 2 Comparison of features used for the concentration level estimation

Collection method	Feature	Tateyama et al. (2004)	Fogarty et al. (2005)	Züger and Fritz (2015)	Tanaka and Fujita (2011)	Proposed
Software	Keyboard input		✓		✓	✓
	Mouse input		✓		✓	✓
	Web communication					✓
	Application switching				✓	
External device	Use of a telephone		✓			
	Opening of a door		✓			
	Eye movements	✓				
	Psycho-physiologic data			✓		

Moreover, instead of using counting-based methods (Fogarty et al. 2005; Tanaka and Fujita 2011), the proposed system records detailed information, such as the application name and hostname along with the type of each operation, as shown in Table 1. This approach not only solves the issue explained in Sect. 2.4 but also can use more information than if only the number of operations was used. For example, clickings in Firefox and Eclipse are regarded as different operations in this method, whereas the counting-based methods treat clickings in Firefox and Eclipse equally. Simultaneously, the importance of each operation is automatically calculated by a machine learning algorithm.

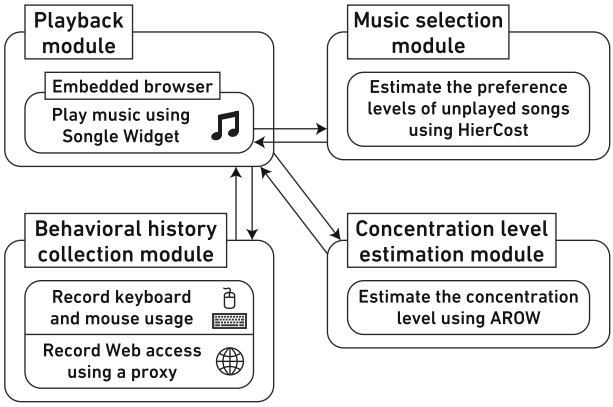

Fig. 5 Implemented modules of FocusMusicRecommender. The user interface is handled by the playback module, and the other three modules operate in the background

Then, the system applies a machine learning algorithm to the n-grams of the hash value of each operation. This was inspired by malware detection methods in which a series of system calls is provided to a learning algorithm in the form of n-grams of their hash values (Rieck et al. 2011; Canali et al. 2012). In particular, it is known that an n-gram performs well in the feature extraction from such sequential data despite its ease of calculation.

4 Implementation

Based on the design described in Sect. 3, we implemented FocusMusicRecommender. In this section, we explain the detail of the implementation, which consists of four modules: behavioral history collection, concentration level estimation, music selection, and playback, as shown in Fig. 5.

4.1 Behavioral history collection

This module records three types of behavioral history as described in Sect. 3.4. The keyboard and mouse input is collected using an API of the operating system, and the Web communication is observed using a local proxy server. The collected data are gathered for each song and sent to the concentration level estimation module.

4.2 Concentration level estimation

This module estimates the concentration level from the user's behavioral history using an online learning algorithm. Here, we use AROW (Crammer et al. 2013) for the following reasons:

– First, it converges quickly because it employs passive-aggressive updating algorithm (Crammer et al. 2006), which makes it suitable for cases where collecting large amounts of labeled data is difficult.
– Second, it tolerates sparse features because it employs confidence weighting (Dredze et al. 2008) that considers the frequency of the features.
– Additionally, it is robust to noise in the labeled data, such as fluctuation of the concentration level given by users, because it assumes a mistake bound that allows some misclassified data considering the presence of the noise.

Every time the song is switched, this module estimates the concentration level from the behavioral history collected during the playback of the last song and sends it to the music selection module. More specifically, it first calculates the n-grams of the behavioral history (here, we set $n = 2$ based on preliminary observations), as described in Sect. 3.4. It then counts the occurrence of each 2-gram and provides the occurrence information to AROW as an input feature vector. Finally, the AROW estimates the concentration level based on the similarity of the occurrence information to those in annotated training data that suggest the relationships between the previously observed behavioral histories and their corresponding concentration levels.

4.3 Music selection

This module then estimates the preference levels of unplayed songs based on those of played songs and their musical features, as mentioned in Sect. 3.3. It applies HierCost (Charuvaka and Rangwala 2015) for the user's preference information, which is obtained by the method described in Sect. 3.2. HierCost is used here because its implementation has been published and it is designed to work well with unbalanced data. Even when the number of songs played is low, the ability of HierCost to handle unbalanced data can reduce the possibility of playing songs the user likes or dislikes very much, which interferes with the user's concentration.

For the calculation of the musical features, the proposed system does not rely on a specific approach, but here we used the same approach as Songrium (Hamasaki and Goto 2013). First, MARSYAS (Tzanetakis and Cook 2000) was used to obtain a 35-dimensional feature vector for each song. The vector consists of the mean and variance of average values of mel-frequency cepstral coefficients calculated across the entire song (26 dimensions), the mean and variance of local spectral features (centroid, rolloff, flux, and zero-crossings) across the entire song (8 dimensions), and the tempo in the chorus section (1 dimension). Then, the first through the third principal components were retained by applying principal component analysis to the feature vector for dimensionality reduction. Lastly, the retained three-dimensional vectors were provided to the HierCost model as input data.

Additionally, as described in Sect. 3.3, the system changes the criterion when selecting the next song among multiple candidates by considering the estimated concentration level. In detail, whereas some recommendation methods (Cardoso et al. 2016; Ikeda et al. 2016) consider the recently played songs, the proposed system also considers the current concentration level (Fig. 6) as follows:

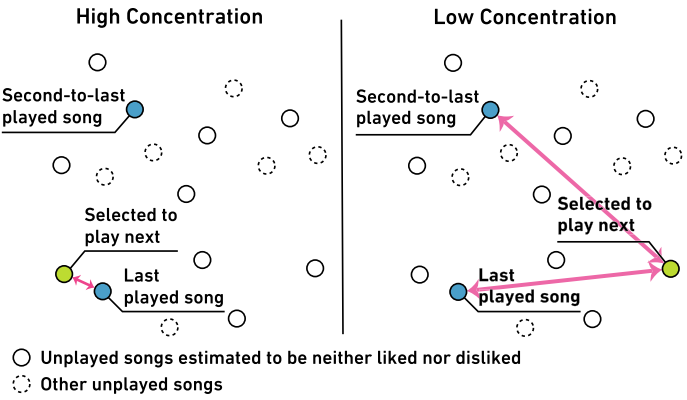

Fig. 6 Conceptual diagram representing the criterion of the song selection in the music similarity space. When the concentration level is high, a song that is the most similar to the last played song is selected. Conversely, when it is low, a song that is the least similar to the previous two songs is selected

Step 1 At first, it randomly selects the first and second songs to start making a listening history.
Step 2 It then estimates the preference levels of unplayed songs and lists all songs labeled as "neither like nor dislike" as candidates. If there is no song labeled as "neither like nor dislike," it uses songs in the order of "like," "dislike," "like very much," and "dislike very much" considering the effect on the concentration level described in Huang and Shih (2011).
Step 3 From the candidates, it selects the third and the following songs based on musical similarities like Pampalk et al. (2005b) as follows:

(a) If the user's concentration level is high, it selects the song with the maximum similarity to the song played immediately before to avoid making sudden changes that can distract the user.
(b) If the user's concentration level is low, it selects the song whose sum of similarities to the last played song and the second-to-last played song is the smallest to give the user an opportunity to change their mood (Wells 1990). It should be noted that this selection is not only based on the song played immediately before, but on the two previous songs. If it selected the song having the least similarity to the last played song, two different genres could be selected alternately, and this would reduce the diversity of the songs played.

Step 4 It goes back to 2.

Here, the music similarity between two songs is measured by the Euclidean distance between their three-dimensional feature vectors.

4.4 Playback

This module plays the song selected by the music selection module using Songle Widget (Goto et al. 2015), which is a framework offering an embeddable music player.

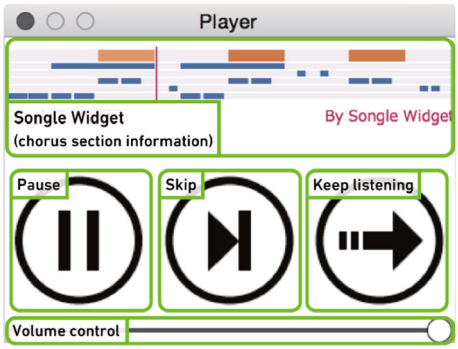

Fig. 7 User interface of FocusMusicRecommender. The chorus section information is shown in the upper half and the controls are shown in the bottom half

The advantage of Songle Widget is that it makes the playback control based on the chorus section easy by exploiting the music structure information estimated automatically on Songle (Goto et al. 2011). Songle is a Web service that provides not only the visualized results of automatic analysis of songs uploaded on the Internet but also a Web-based interface that enables users to correct errors in the automatic analysis. Thus, we can also expect Songle Widget to deliver accurate user-corrected chorus information to the module.

The module also handles the user interface (Fig. 7). In the upper half, the chorus section information of the playing song is shown. In the bottom half, a slider to adjust volume and three buttons to pause, skip, and keep listening are shown.

5 Preliminary experiments

To confirm the validity and effectiveness of our design strategy, we first conducted three preliminary experiments, which focus on specific components of the proposed system. We started by verifying the accuracy of the concentration level estimation described in Sect. 4 by collecting annotated data. We then evaluated the validity of the preference level determination described in Sect. 3.2. We also confirmed the generality of the playback function used in the preference level determination through a Web-based experiment involving crowd workers. Here, we show the detailed procedures and results of those experiments.

5.1 Data preparation

5.1.1 Songs

For the experiments, we first constructed a set of songs to be presented to participants. We used the top 50 most frequently played songs with the tag "VOCALOID," which are songs created using a singing synthesis software, in the popular Japanese video-sharing service *niconico* (http://nicovideo.jp/). We used those songs because we had confirmed that accurate user-corrected chorus section information for all of them was

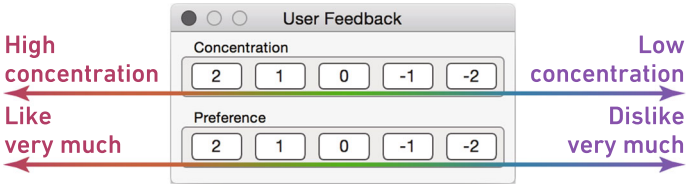

Fig. 8 Dialog box presented to the participants for entering their concentration and preference levels. This dialog box was presented whenever a song was switched

available via Songle Widget. Furthermore, those songs tend to cover diverse genres[2] as they are created in a user-generated content community on a voluntary basis (Hamasaki et al. 2008; Hamasaki and Goto 2013). In fact, 15 different tags indicating the genres are used for the 50 songs, such as "*VOCAROCK* (used for rock music)," "*Vocaloid Japanese-style music*," and "*Mikuno-Pop* (used for electro-pop music)."

5.1.2 Annotated behavioral data

For the experiments, we also collected system users' behavioral information during the playback of the songs along with their subjective concentration and preference levels. In detail, eight students (17 to 24 years old) who were fluent in Japanese and in the habit of listening to music while working on a personal computer participated voluntarily. Using headphones and in a quiet room, they listened to all songs in the constructed set in random order while being presented with "keep listening" and "skip" buttons.

During this procedure, four participants wrote new documents using word processing applications and the other four worked on programming on a personal computer. They were instructed to work in the same way as usual without being given specific tasks to complete, and they were not interrupted before they had finished listening to the 50 songs. Their input and Web communication histories were collected automatically, and they were informed beforehand that the collected data are stored in the form of hashes to preserve their privacy so that they can work as they usually do.

Simultaneously, we collected their concentration and preference levels using a dialog box (Fig. 8), which is presented each time a song was played. Our dialog box had five-point scales ranging from "like very much" to "dislike very much" for preference level and from "high concentration" to "low concentration" for concentration level. We acknowledge that asking the participants to enter their concentration levels with a dialog box can influence their concentration levels; however, we followed previous studies of concentration level estimation (Fogarty et al. 2005; Tanaka and Fujita 2011) to make the results comparable. We note that this is only for preliminary experiments, and FocusMusicRecommender does not require such explicit feedback.

[2] As of October 1, 2017, 141 tags indicating the user-defined music genre of VOCALOID songs are listed in "*nicopedia*," a Wiki system for topics related to *niconico* (http://dic.nicovideo.jp/id/252926 in Japanese).

Table 3 Confusion matrix of the concentration level estimation (five-class)

Estimated label		User-entered label				
		2	1	0	−1	−2
High	2	**28**	18	18	9	5
	1	14	**39**	12	11	9
	0	11	10	**22**	12	12
	−1	5	11	15	**30**	28
Low	−2	5	12	11	22	**31**
Total		63	90	78	84	85

The diagonal components of the confusion matrix (i.e., the number of times that the user-entered label was correctly estimated) are highlighted in bold

Table 4 Confusion matrix of the concentration level estimation (two-class)

Estimated label	User-entered label	
	High (2, 1)	Low (0 ∼ −2)
High concentration	**99**	64
Low concentration	54	**183**
Total	153	247

The diagonal components of the confusion matrix (i.e., the number of times that the user-entered label was correctly estimated) are highlighted in bold

5.2 Accuracy of concentration level estimation

We verified the correspondence of the estimated concentration level and the validation data that the participants entered by five-fold cross-validation where instances of the participants were distributed randomly across the folds in the same manner as Züger and Fritz (2015) did. The confusion matrices in the five- and two-class estimation are presented in Tables 3 and 4. We found that the accuracy was 37.5% in the five-class estimation and 70.5% in the two-class estimation, while the F1 score was 37.7% in the five-class estimation and 62.7% in the two-class estimation.

Though the results came from different studies, the accuracy of the two-class estimation was comparable to that of the previous methods we mentioned in Sect. 2.4. More specifically, the F1 score of the proposed method (62.7%) was lower than that of the previous methods that use external sensors; Fogarty et al. (2005) achieved 70.5% using a physical sensor, and Züger and Fritz (2015) achieved 69.7% using biometric sensors. However, it was higher than that of Tanaka and Fujita (2011), which achieved 38.8% using features that can be collected without external sensors.[3] We note that, in calculating the F1 scores of the previous methods (Fogarty et al. 2005; Tanaka and Fujita 2011) and the proposed method, we converted the multiclass estimation results into two-class estimation results by following the procedure of Züger and Fritz (2015). That is, a concentration level labeled "neither high nor low" (= 0) is categorized as

[3] The confusion matrix of the method using eye movements is not presented in Tateyama et al. (2004).

Table 5 F1 scores of concentration level estimated with and without the use of Web communication history

		Without Web communication	With Web communication
Features	Keyboard input	✓	✓
	Mouse input	✓	✓
	Web communication		✓
F1 score	5-class[a]	32.5%	**37.3%**
	2-class	52.8%	**62.7%**

[a] We calculated the macro average across five classes
In each row for F1 score, the higher accuracy is highlighted in bold

"low." The results confirm the effectiveness of the proposed method for estimating the concentration level.

We also compared the F1 scores of the concentration levels estimated from behavioral history with and without consideration of Web communication history. The results are listed in Table 5, which indicates that using the Web communication history improves the accuracy of the estimation.

Furthermore, the effectiveness of the Web communication history is also confirmed from the top 10 most important features for the estimation (Table 6). They are ranked based on the mean vector $\boldsymbol{\mu}$ of AROW (Crammer et al. 2013), which has a role similar to that of the weight vector in linear classification and shows the importance of the corresponding feature in the estimation. Table 6 shows that accesses to social networking services, such as Twitter and Facebook, are significant clues (Rank #1, #3, #4, and #8 in Table 6) for estimating the concentration level. Additionally, an interesting point in the result is that repeated pressing of the backspace key plays an important role in the estimation (Rank #2, #5, and #10 in Table 6). Such repetition can occur frequently when a user is editing a considerable length of text in the applications listed in the rows, and thus, it would be associated with the high level of the user's concentration.

5.3 Validity of preference level determination

To confirm whether the proposed method can determine a user's preference level from their implicit feedback and estimated concentration level (Sect. 3.2.1), we evaluated the coherency between the preference level determined using the proposed method and that entered by the participants. We first checked whether the preference level of songs the participants wanted to keep listening to was high and that of songs they wanted to skip was low. The result is shown in Table 7. Here, Spearman's correlation coefficient of the preference level the participants entered to the user feedback is 0.62 ($p < 0.01$). Note that the p-value here indicates the probability that the data would have arisen even if there were no actual correlation.

We then evaluated whether the preference level determined in combination with the estimated concentration level appropriately reflects that entered by the participants. The result is shown in Table 8, and Spearman's correlation coefficient between the

Table 6 Top 10 most important features for the concentration level estimation (two-class). Each row represents two consecutive operations (e.g., the row of rank #1 means that the user accessed twitter.com immediately after clicking in Google Chrome), and the sign next to the rank number represents whether the operations were associated with high (+) or low concentration (−). They suggest that the Web communication history is an important clue for estimating the concentration level

Rank	2-gram features of two consecutive operations		
1 (−)	"mouse Google Chrome 1"[a]	→	"net GET twitter.com"
2 (+)	"key Terminal <[Backspace]>"	→	"key Terminal <[Backspace]>"
3 (−)	"net GET scontent.xx.fbcdn.net"[b]	→	"net GET scontent.xx.fbcdn.net"
4 (−)	"mouse Google Chrome 1"	→	"net GET pbs.twimg.com"[c]
5 (+)	"key Microsoft Word <[Backspace]>"	→	"key Microsoft Word <[Backspace]>"
6 (+)	"net GET mail.google.com"	→	"net GET 0.docs.google.com"
7 (−)	"mouse Google Chrome 5"[d]	→	"mouse Google Chrome 5"
8 (−)	"net GET www.facebook.com"	→	"net GET scontent.xx.fbcdn.net"
9 (+)	"mouse Google Chrome 7"[e]	→	"mouse Google Chrome 7"
10 (+)	"key Xcode <[Backspace]>"	→	"key Xcode <[Backspace]>"

[a] It means clicking a left button in Google Chrome
[b] It means connecting a server that serves static contents of Facebook
[c] It means connecting a server that serves images on Twitter
[d] It means scrolling down in Google Chrome
[e] It means scrolling right (i.e., a gesture to go back) in Google Chrome

Table 7 Correspondence of the "keep listening" and "skip" feedback to the user-entered preference level ($\rho = 0.62$, $p < 0.01$)

User feedback	Preference level			Total
	Like (2, 1)	Neither like nor dislike (0)	Dislike (−1, −2)	
Keep listening	**84**	15	10	109
Do nothing	58	**107**	33	198
Skip	8	4	**81**	93
Total	150	126	124	400

The diagonal components of the confusion matrix (i.e., the number of times that the user's preference level and their feedback matched our hypothesis) are highlighted in bold

preference level determined by the proposed method and the one that the participants entered is 0.66 ($p < 0.01$). Therefore, the hypothesis mentioned in Sect. 3.2.2 was supported, and the employment of the proposed determination method to acquire the preference information without burdening the user can be justified.

5.4 Generality of summarized playback

As mentioned in Sect. 3.1, the proposed system introduced the playback function that automatically summarizes songs based on the chorus section information to enable implicit feedback using the "keep listening" button. While its effectiveness in determining the user's preference level was confirmed in Sect. 5.3, it is known that the

Table 8 Correspondence of the combination of the user feedback and estimated concentration level to the user-entered preference level ($\rho = 0.66$, $p < 0.01$)

User feedback (Concentration level)	Preference level					Total
	Like			Dislike		
	2	1	0	−1	−2	
Keep listening (High concentration)	**22**	8	3	2	0	35
Keep listening (Low concentration)	21	**33**	12	6	2	74
Do nothing	13	45	**107**	27	6	198
Skip (Low concentration)	1	5	3	**35**	19	63
Skip (High concentration)	0	2	1	6	**21**	30
Total	57	93	126	76	48	400

The diagonal components of the confusion matrix (i.e., the number of times that the user's preference level and the combination of their feedback and the estimated concentration level matched our hypothesis) are highlighted in bold

adaptation of such playback function can be subjected to the music culture or the usual listening habit of the user (Bull 2006). Thus, although summarized playback has been implemented in consumer products (see Sect. 3.2.1), it is desirable to test the function with participants from various backgrounds to ensure the generality of the proposed method.

In that respect, evaluation involving crowd workers is one of the popular methods for comparing the usage of interfaces among diverse participants (Liu et al. 2012). Fortunately, whereas providing the complete implementation of the proposed system to crowd workers is difficult due to its dependency on the operating system API, as described in Sect. 4.1, the playback interface without the recommendation function can be replicated on the Web. That is, crowd workers can try the summarized playback with the "keep listening" button and evaluate its usage, even though the songs are presented through a random shuffle.

For this purpose, we conducted an experiment by recruiting 25 participants from Asia, North America, and South America on Amazon Mechanical Turk. The participants used the interface while being presented with the "keep listening" and "skip" buttons and listened to the songs in a random order. Here, because most songs in the ones we prepared in Sect. 5.1.1 were Japanese lyrics, presenting the songs to crowd workers would confuse them and yield a less grounded result. Instead, we collected 20 English-lyrics most-played VOCALOID songs to preserve the composition of the songs to be presented. After at least 15 minutes of use while working, they were directed to a survey asking their opinions on the summarized playback compared with the continuous playback in regular players using a five-point scale of "much better," "better," "not so different," "worse," and "much worse" along with their remark on the "keep listening" button. To ensure the validity of the results, we implemented the interface so that the participants could not proceed to the survey without spending at least 15 minutes with the interface, and we manually inspected their playback history, such as confirming not skipping all songs immediately.

All the participants responded positively about the summarized playback: 15 participants (60.0%) selected "much better," 10 participants (40.0%) selected "better," and

Table 9 Distribution of the feedback given to the proposed interface by the participants described in Sect. 5.1.2 and the participating crowd workers. The p-value of the chi-square test for homogeneity suggests that there is no significant difference in the usage of the interface between them regarding both the "keep listening" and "skip" buttons

	User Feedback			Total
	Keep listening	Do nothing	Skip	
Table 7	109	198	93	400
Crowd workers	63	160	53	276
p-value	0.301	–	0.245	–

none selected "not so different," "worse," or "much worse." Most of them commented affirmatively about the "keep listening" button, such as "sometimes I want to listen to the full song so its useful to me," whereas some participants commented "I didn't use that button because I prefer the mixed playback." One suggested an improvement of the interface design: "I think it would be better if it gave me some visual feedback that it was pressed."

Moreover, as shown in Table 9, their playback history on the usage of the "keep listening" and "skip" buttons showed a similar distribution to that shown by the participants described in Sect. 5.1.2. From Pearson's chi-square test of homogeneity across the participants and participating crowd workers, we obtained a p-value of 0.301 for pressing the "keep listening" button and 0.245 for the "skip" button. In other words, we could not find a significant difference in the usage of the summarized playback with the "keep listening" button regarding the backgrounds of the participants.

In sum, these results from diverse participants suggested the generality of the usage of our playback function that automatically summarizes songs while providing the "keep listening" button. Thus, along with the results in Sect. 5.3, it is implied that the proposed method is a novel approach for determining the user's preference level for played songs without burdening the user.

6 User study

To confirm the performance and effectiveness of the entire function of the proposed method as a recommendation system, we then conducted a user study with several comparison implementations. We first evaluated its recommendation performance by comparing the results obtained using implementations with different recommendation strategies and then evaluated its effectiveness by comparing the results obtained using implementations of the proposed version with different prioritization. We also examined the comments obtained from the participants.

6.1 Procedure

This user study involved the same eight participants in the experiments described in Sect. 5.1.2. Each participant experienced four recommendation systems consisting of FocusMusicRecommender and the comparison implementations described below.

Baseline (BL)

As a baseline, we implemented a recommendation system based on the method proposed by Pampalk et al. (2005b), which uses the skip operation as feedback. We modified the selection procedure explained in Sect. 2.3 so as to play songs dissimilar to songs that the user pressed either the "skip" or "keep listening" button, instead of avoiding songs similar to songs the user skipped. In other words, the binary feedback of Pampalk et al. (2005b) shown in Fig. 3 (A) cannot avoid playing songs the user may like very much and would not yield a result comparable to that of other implementations. Thus, it was extended to the three levels shown in Fig. 3(B) for avoiding songs the user may like or dislike very much. Consequently, this system is expected to play songs that the user might neither like nor dislike by avoiding songs that the user might press either the "skip" or "keep listening" button in an online learning manner.

FocusMusicRecommender Not Considering Concentration Level (FMR-1)

We implemented a simplified version of FocusMusicRecommender, which does not consider the user's concentration level. This system uses the feedback shown in Fig. 3(B), and thus, the preference levels of unplayed songs are estimated in the three levels. In this case, the estimation process is expected to behave like regular classifiers that do not use the hierarchical information. Additionally, since the selection process described in Sect. 4.3 depends on the concentration level, the system instead selects the next song randomly from unplayed songs that are estimated to be neither liked nor disliked.

FocusMusicRecommender Considering Concentration Level (FMR-2)

We of course implemented FocusMusicRecommender in the proposed version. This system is based on the five-level preference refined using the estimated concentration level, as shown in Fig. 3(C), and changes the criterion for selecting the next song according to the concentration level, as explained in Sect. 4.3.

FocusMusicRecommender Playing Songs the User May Like Very Much (FMR-3)

We additionally implemented a modified version of FocusMusicRecommender, which plays songs the user may like very much in the same way as conventional recommendation systems. The system uses the same preference determination and song selection method as FMR-2 but changes the priority for listing candidates from that described in Sect. 4.3 to the order of "like very much," "like," "neither like nor dislike," "dislike," "dislike very much." This is intended to confirm the influence of playing songs the user may like very much on interfering with the user's concentration by comparing its usage with that of FMR-2.

The participants used each implementation until 30 of the 50 songs mentioned in Sect. 5.1.1 have been played (i.e., they listened to 120 songs in total). The conditions of work contents and environments were inherited from Sect. 5.1.2. Specifically, four participants worked on document writing using word processing applications, and the other four worked on programming tasks. When they completed the use of FMR-2, we conducted short interviews with them asking them to comment on their impressions of the songs played (i.e., asking "how do you feel about the played songs?") and the

ease of use (i.e., asking "how do you feel about the usage of the provided system?"), which lasted approximately 10–15 minutes.

We note that the data collection described in Sect. 5.1.2 was performed at least one month before the user study commenced to avoid anomalous results due to short-term preference changes caused by repeated listening. Additionally, though the collected behavioral history was used to train the concentration level estimation module of the FMR-2 and FMR-3, the collected preference information was not used in any of the implementations. Furthermore, each implementation was used on separate days to avoid the effect caused by fatigue, as the experience could take up to an hour.

6.2 Comparison of recommendation performances

Table 10 shows the distribution of the preference level the participants entered beforehand during the data collection in Sect. 5.1.2 and the number of operations the participants performed for songs played by BL, FMR-1, and FMR-2. Compared with the distribution of the population shown in the bottom row of Table 8, the hypergeometric p-value of avoiding songs liked very much or disliked very much is 0.1×10^{-1} in BL, 2.4×10^{-7} in FMR-1, and 1.6×10^{-13} in FMR-2. Here, the hypergeometric p-value indicates how the distribution of the preference levels of the songs chosen by each system was different from that in the case that the songs were drawn by random sampling. Thus, the results suggest that the use of any of the three implementations can significantly reduce the playback of songs liked or disliked very much (i.e., songs that may interfere with the user's concentration), compared to playing songs by a random shuffle. Additionally, it is demonstrated that FMR-2, which implements the proposed method, played fewer songs that would decrease the concentration level than BL and FMR-1; thus, it is suitable for use during work. This is supported by the fact that FMR-2 caused fewer interruptions due to pressing the "skip" or "keep listening" button than the BL and FMR-1.

The difference between the BL and FMR-1 results can be explained by the use of the machine learning algorithm. That is, FMR-1 decides which song to play based on the preference levels of unplayed songs estimated by the learning algorithm, whereas

Table 10 Distribution of the preference levels of the songs played by BL, FMR-1, and FMR-2 along with the number of operations recorded. While all of them played fewer songs that the participants liked very much or disliked very much (i.e., songs that can decrease their concentration level) compared to a random shuffle, the proposed method (FMR-2) played fewest

	Preference level					Participants pressed		Total number of operations
	Like			Dislike		Keep listening	Skip	
	2	1	0	−1	−2			
BL	43	62	79	38	18	58	44	102
FMR-1	30	45	111	34	20	49	42	91
FMR-2	22	48	121	38	11	38	32	**70**

The smallest value of the total number of operations, which can be associated with the number of caused interruptions, is highlighted in bold

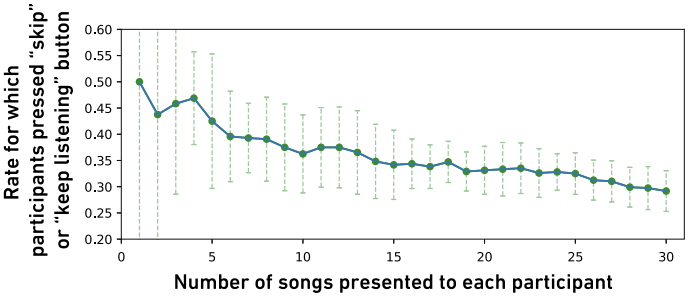

Fig. 9 Transition of the rate of songs played by FMR-2 for which participants pressed the "skip" or "keep listening" button. Error bars represent the standard error of the mean. The decrease in this rate suggests that the proposed method can adapt to a new user as a certain number of songs are played

BL bases the decision on the neighbors of each song. On the other hand, the difference between FMR-1 and FMR-2 is due to the precise determination of the preference level as described in Sect. 3.2.2. It enables FMR-2 not only to estimate the preference levels of unplayed songs precisely but also to prioritize the levels in accordance with the effect on the concentration level by combining with the hierarchical classification algorithm. In fact, FMR-2 played fewer liked very much and disliked very much songs than FMR-1 even though it played more liked and disliked songs, resulting in fewer interruptions, as shown in Table 10.

Additionally, Fig. 9 shows that the rate of songs for which the participants pressed the "skip" or "keep listening" button dropped as the number of played songs increased and stabilized after about 10 songs. This implies that, though further investigations are needed, the user can benefit from the proposed system with the playback of few songs before getting disappointed due to the less accurate results when the user uses the system for the first time. Here, we would like to note that this adaptation ability is attributable to HierCost, the classification algorithm used in FMR-2, as it works well with small and unbalanced data.

6.3 Comparison of effects on the user

Table 11 shows the distribution of the preference level and number of operations for songs played by FMR-2, and FMR-3. The difference in the distribution of the preference level reflects the difference in their recommendation priority: FMR-2 plays songs the user may neither like nor dislike, while FMR-3 plays songs the user may like very much. Playing songs the user may neither like nor dislike results in a 32.7% reduction in the number of operations the participants performed for played songs even though FMR-2 and FMR-3 use the same preference determination and song selection method. This implies that listening to songs one may neither like nor dislike helps one concentrate on one's work (Huang and Shih 2011), as stated in Sect. 1.

The effectiveness of the proposed method is also supported by Fig. 10. It shows how the rate of liked or disliked songs for which the participants pressed the "skip" or "keep listening" button changed over time. One infers from the hypothesis presented in

Table 11 Distribution of the preference levels of the songs played by FMR-2 and FMR-3 along with the number of operations recorded. Playing songs the user may neither like nor dislike significantly reduces the number of interruptions

	Preference level					Participants pressed		Total number of operations
	Like			Dislike		Keep listening	Skip	
	2	1	0	−1	−2			
FMR-2	22	48	121	38	11	38	32	**70**
FMR-3	49	69	84	25	13	67	37	104

The smallest value of the total number of operations is highlighted in bold

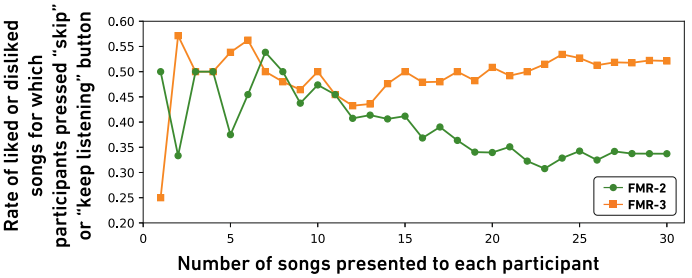

Fig. 10 Transition of the rate of liked or disliked songs played by FMR-2 and FMR-3 for which participants pressed the "skip" or "keep listening" button. The decrease in this rate of FMR-2 in comparison with that of FMR-3 implies that the proposed method helped the users to concentrate

Sect. 3.2.2 and the results shown in Sect. 5.3 that the user is unlikely to press the "skip" or "keep listening" button for liked or disliked songs when the user's concentration level is high. In that respect, compared to FMR-3, Fig. 10 indicates that the participants got concentrated as they made use of FMR-2.

Besides the effectiveness of FMR-2 in helping the user concentrate, these results also suggested the effectiveness of FMR-3 as a regular recommendation system that is intended to meet the user's preference. Specifically, Table 11 shows that FMR-3 played much more songs liked very much, which resulted in increasing the number of the use of the "keep listening" button. Since our preference determination method, which is also used in FMR-3, does not require users to input preference information explicitly, providing FMR-3 to users who are not working would also be beneficial.

6.4 User comments

After using FMR-2, the participants provided positive comments about the songs played, e.g., "I think they were good for concentrating," "they were good choices," "they were moderately suitable for working," "I was able to work comfortably while listening," "although they matched my preference, they never got in the way of working," "I was bothered neither by music nor by my surroundings," "I paid less attention to music than usual," and "I think they became more suitable for working as I made use of the system." Comments such as "I didn't feel it burdened me," "I didn't particu-

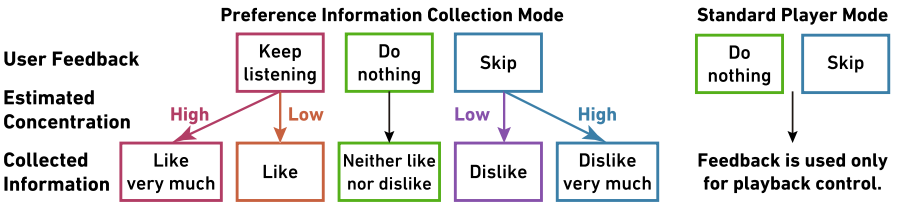

Fig. 11 Overview of the standard player mode. FocusMusicRecommender can be used with the same interface as standard music players once the preference information is collected

larly mind it," and "I didn't feel uncomfortable" were made about the "keep listening" button.

One participant mentioned that using the "keep listening" button seemed to interrupt work more than using the "skip" button. This is probably due to his unfamiliarity caused by the fact that a "skip" button is widely used by many music players, while a "keep listening" button is not. He also said that "using a 'keep listening' button is much easier than using a precise scale and entering a preference level for each song," and therefore the efficiency of the feedback method is supported. Additionally, we remark that the number of button operations is expected to decrease if the user continues to use a "keep listening" button, and as shown in Fig. 9, this will result in fewer interruptions.

Furthermore, once the preference information of 10 or 20 songs is accumulated, it is possible for the user to disable the automatic summarization function and use the system with the same interface as standard music players (Fig. 11). In this case, the user feedback is used only for playback control and not used for recommendation. When the user's preference changes (e.g., songs not estimated to be disliked very much are skipped many times), the preference information collection could be reactivated.

7 Limitations

While our experiments (Sect. 5) and user study (Sect. 6) suggested the effectiveness of FocusMusicRecommender, there are some limitations. First, to certify the effectiveness of the proposed method, it is desirable to measure the concentration level of participants who are using FocusMusicRecommender and evaluate how songs played affect their concentration level, rather than discussing the effect based on observable data (e.g., the number of operations), as we did in Sect. 6. However, measuring the concentration level without involving expensive equipment, such as fMRI, requires further consideration. One feasible option is the use of psycho-physiological sensors, as Züger and Fritz (2015) did; but such psycho-physiological data do not fully reflect participants' internal status, as Züger and Fritz (2015) gave an accuracy of 69.7% in estimating participants' self-reported interruptibility. Another option is measuring the score of an attention test, as Huang and Shih (2011) did; however, it makes participants' situations artificial, which would be apart from the actual situation where FocusMusicRecommender is used.

Second, the number of participants was relatively small to ensure the generalizability of the results. This is partially due to the implementation of FocusMusicRecommender, which depends on the API of the operating system to collect a user's behavioral

history and make it difficult to conduct the experiments on a large scale. Instead, we conducted the experiment to evaluate a part of FocusMusicRecommender that is independent of the API, i.e., the usage of its summarized playback interface, involving crowd workers in Sect. 5.4. Still, there is room for expansion in terms of the number of participants to ensure its generality to users from various cultural backgrounds.

In this context, our selection of the songs used in the experiments and user study could bring bias with respect to the participants' perceptions and usages. As explained in Sect. 5.1.1, we selected "VOCALOID" songs because they can balance the requirement for accurate chorus section information and the coverage of diverse genres. Therefore, additional investigations involving different sets of songs and a greater number of participants are desirable to ensure the generalizability of our results.

Furthermore, the accuracy of the concentration level estimation can be affected by conditions of the experiments, since it would highly depend on the work content. As presented in Sect. 3.4, the proposed system exploits the user's detailed behavioral history, which reflects the work content, for the estimation. Thus, when we train the concentration level estimation module using the data obtained in experiments like those described in Sect. 5.1.2, the accuracy during completely different tasks (e.g., graphic design) will get worse.

Meanwhile, this problem can be easily addressed by acquiring a small amount of additional training data from the user in the same manner as presenting the dialog box (Fig. 8) to the user in the experiments. This is because, as mentioned in Sect. 4.2, the proposed system uses an online learning algorithm, AROW (Crammer et al. 2013), and can adapt to the new data with a small computational cost. Moreover, by introducing the framework of active learning, it is possible to reduce the number of inquiries to the user, which results in the suppression of the burden on the user. In detail, by asking the user to enter the current concentration level only when the confidence of the estimation is low, the proposed system can adapt to new data efficiently (Lu et al. 2016).

Regarding the concentration level estimation, there is concern about the privacy of the user because the estimation relies on detailed behavioral history. This can also be addressed by taking advantage of the learning algorithm used in the proposed system. That is, the computational cost of the estimation is independent of the amount of the training data. In other words, it is a lightweight algorithm in which the estimation can be performed in the user's computer without sending the user's behavioral history to the outside world.

Moreover, it is pointed out that user preferences naturally evolve over time (Gama et al. 2014). Although in the proposed method, there is no consideration for the change of preference, it can be easily dealt with by introducing a sliding window, one of the popular approaches to adapting to the transition (Gama et al. 2014). In detail, by considering only a fixed amount of the latest playback history in the preference level estimation for unplayed songs (Sect. 4.3), the recommendation by the system can be based on recent data reflecting the change of the preference.

8 Conclusion and future work

We have presented FocusMusicRecommender, a music recommendation system designed to help people concentrate while working on personal computers. Based on previous studies on the effect of background music on the concentration, the system gives priority to songs that a user may neither like nor dislike rather than pursuing songs that a user may like very much as conventional systems do. To be used while working, it automatically selects such songs according to the user's situation without asking the user to input preference information explicitly.

To realize the proposed system, we first introduced a feedback method that obtains three levels of preference "like very much," "neither like nor dislike," and "dislike very much" while suppressing the burden on the user using a "keep listening" and "skip" button. We then introduced a process refining the preference level by determining the degree of "like" or "dislike" according to the user's concentration level that is automatically estimated. Furthermore, we proposed a method for estimating preference levels of unplayed songs and selecting the most suitable song by considering the relationship between the concentration level and preference levels.

The results of our experiments and user study confirmed the validity and effectiveness of the proposed method as well as the suitability of the recommended songs. The experiments also confirmed the effectiveness of the proposed method for estimating the concentration level from the user's behavioral history collected without using external sensors. They showed that its estimation accuracy was better than those of the previous methods described in Sect. 2.4.

8.1 Future Work

For future work, we would like to experiment with alternative approaches and more participants. For example, we are exploring a different music selection method that considers the novelty and diversity of the recommended songs because they are sometimes considered in the quality assessment of recommendation systems. As familiarity with songs is known to affect the preference level (Brattico and Pearce 2013), using a recommendation algorithm that controls the familiarity would be another interesting approach for designing a recommendation system to be used while working.

Additionally, FocusMusicRecommender can be implemented to incorporate other learning algorithms. For instance, reinforcement learning is often employed in recommendation systems to explore various candidates in order to find one meeting a user's preference (Moling et al. 2012; Wang 2020). Such an algorithm can perform better than our implementation of music selection (Sect. 4.3) in terms of providing the user a chance to change their mood by playing various songs. More generally, leveraging the state-of-the-art methods (Zhang et al. 2019; Deldjoo et al. 2020) for designing recommendation systems to be used while working would be a fruitful direction.

Furthermore, exploring the use of other musical features remains future work. Since the proposed system depends on the musical similarity rather than its calculation procedure, as mentioned in Sect. 4.3, other methods like a hybrid approach that combines acoustic features and related information from Web pages (Knees et al. 2007; Taka-

hashi et al. 2008) can also be used. Specifically, given that the system plays songs in an abridged manner, we want to compare the result with that obtained using a feature extraction method that considers the structure and variation in the song. For example, Deldjoo et al. (2019) enabled extracting features from movie trailers coherent with the corresponding full-length movies by incorporating musical features that are similar to ours and i-vector features (Eghbal-zadeh et al. 2015), which can be applied to the proposed system.

We would also like to explore new interactions that leverage the estimated concentration level. For instance, FocusMusicRecommender can prompt a user to take a break when the estimated concentration level stays low. Additionally, when the user accepts the recommendation of taking a break, the system can help the user change their mood seamlessly through the song selection. That is, the system changes the priority of the selection described in Sect. 4.3 and gradually increases the number of songs the user may like very much during the playback.

References

Adomavicius, G., Tuzhilin, A.: Context-aware recommender systems. In: Ricci F, Rokach L, Shapira B (eds) Recommender Systems Handbook, Springer US, Boston, MA, 191–226, (2015), https://doi.org/10.1007/978-1-4899-7637-6_6

Baltrunas, L., Kaminskas, M., Ludwig, B., Moling, O., Ricci, F., Aydin, A., Lüke, K., Schwaiger, R.: InCarMusic: context-aware music recommendations in a car. In: Proceedings of the 12th International Conference on E-Commerce and Web Technologies, Springer, Berlin, Heidelberg, 89–100, (2011), https://doi.org/10.1007/978-3-642-23014-1_8

Biswas, D., Lund, K., Szocs, C.: Sounds like a healthy retail atmospheric strategy: effects of ambient music and background noise on food sales. J. Acad. Mark. Sci. **47**(1), 37–55 (2018). https://doi.org/10.1007/s11747-018-0583-8

Brattico, E., Pearce, M.: The neuroaesthetics of music. Psychol. Aesthet. Creat. Arts **7**(1), 48–61 (2013). https://doi.org/10.1037/a0031624

Bull, M.: Investigating the culture of mobile listening: from Walkman to iPod. In: O'Hara, K., Brown, B. (eds) Consuming music together: social and collaborative aspects of music consumption technologies, Springer, Dordrecht, Netherlands, 131–149, (2006), https://doi.org/10.1007/1-4020-4097-0_7

Canali, D., Lanzi, A., Balzarotti, D., Kruegel, C., Christodorescu, M., Kirda, E.: A quantitative study of accuracy in system call-based malware detection. In: Proceedings of the 2012 International Symposium on Software Testing and Analysis, ACM, New York, NY, 122–132, (2012), https://doi.org/10.1145/2338965.2336768

Cardoso, J.P.V., Pontello, L.F., Holanda, P.H.F., Guilherme, B., Goussevskaia, O., da Silva, A.P.C.: Mixtape: direction-based navigation in large media collections. In: Proceedings of the 17th International Society for Music Information Retrieval Conference, ISMIR, Montreal, Canada, 454–460, (2016)

Charuvaka, A., Rangwala, H.: HierCost: improving large scale hierarchical classification with cost sensitive learning. In: Proceedings of the 2015 European Conference on Machine Learning and Principles and

Practice of Knowledge Discovery in Databases, Springer, Cham, Switzerland, 675–690, (2015), https://doi.org/10.1007/978-3-319-23528-8_42
Cooper, M.L., Foote, J.: Automatic music summarization via similarity analysis. In: Proceedings of the 3rd International Conference on Music Information Retrieval, ISMIR, Montreal, Canada, 81–85, (2002)
Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. J. Mach. Learn. Res. **7**(19), 551–585 (2006)
Crammer, K., Kulesza, A., Dredze, M.: Adaptive regularization of weight vectors. Mach. Learn. **91**(2), 155–187 (2013). https://doi.org/10.1007/s10994-013-5327-x
Czerwinski, M., Horvitz, E., Wilhite, S.: A diary study of task switching and interruptions. In: Proceedings of the 2004 ACM SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, 175–182, (2004), https://doi.org/10.1145/985692.985715
Dannenberg, R.B., Goto, M.: Music structure analysis from acoustic signals. In: Havelock D, Kuwano S, Vorländer M (eds) Handbook of signal processing in acoustics, Springer, New York, NY, 305–331, (2008), https://doi.org/10.1007/978-0-387-30441-0_21
Deldjoo, Y., Dacrema, M.F., Constantin, M.G., Eghbal-zadeh, H., Cereda, S., Schedl, M., Ionescu, B., Cremonesi, P.: Movie genome: alleviating new item cold start in movie recommendation. User Model. User-Adap. Inter. **29**(2), 291–343 (2019). https://doi.org/10.1007/s11257-019-09221-y
Deldjoo, Y., Schedl, M., Cremonesi, P., Pasi, G.: Recommender systems leveraging multimedia content. ACM Comput. Surv. **53**(5), 106:1-106:38 (2020). https://doi.org/10.1145/3407190
Dredze, M., Crammer, K., Pereira, F.: Confidence-weighted linear classification. In: Proceedings of the 25th International Conference on Machine Learning, JMLR, Cambridge, MA, 264–271, (2008), https://doi.org/10.1145/1390156.1390190
Eghbal-zadeh, H., Schedl, M., Widmer, G.: Timbral modeling for music artist recognition using i-vectors. In: Proceedings of the 23rd European Signal Processing Conference, IEEE, New York, NY, 1286–1290, (2015), https://doi.org/10.1109/EUSIPCO.2015.7362591
Fogarty, J., Hudson, S.E., Atkeson, C.G., Avrahami, D., Forlizzi, J., Kiesler, S.B., Lee, J.C., Yang, J.: Predicting human interruptibility with sensors. ACM Transactions Computer Human Interact. **12**(1), 119–146 (2005). https://doi.org/10.1145/1057237.1057243
Fox, J.G.: Background music and industrial efficiency-a review. Appl. Ergon. **2**(2), 70–73 (1971). https://doi.org/10.1016/0003-6870(71)90072-X
Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. ACM Comput. Surv. **46**(4), 44:1-44:37 (2014). https://doi.org/10.1145/2523813
Goto, M., Yoshii, K., Fujihara, H., Mauch, M., Nakano, T.: Songle: A web service for active music listening improved by user contributions. In: Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR, Montreal, Canada, 311–316, (2011)
Goto, M., Yoshii, K., Nakano, T.: Songle Widget: Making animation and physical devices synchronized with music videos on the web. In: Proceedings of the 2015 IEEE International Symposium on Multimedia, IEEE, New York, NY, 85–88, (2015), https://doi.org/10.1109/ISM.2015.64
Hallam, S., Price, J., Katsarou, G.: The effects of background music on primary school pupils' task performance. Educ. Stud. **28**(2), 111–122 (2002). https://doi.org/10.1080/03055690220124551
Hamasaki, M., Goto, M.: Songrium: A music browsing assistance service based on visualization of massive open collaboration within music content creation community. In: Proceedings of the 9th International Symposium on Open Collaboration, ACM, New York, NY, 4:1–4:10, (2013), https://doi.org/10.1145/2491055.2491059
Hamasaki, M., Takeda, H., Nishimura, T (2008) Network analysis of massively collaborative creation of multimedia contents: case study of Hatsune Miku videos on Nico Nico Douga. In: Proceeding of the 1st International Conference on Designing Interactive User Experiences for TV and Video, ACM, New York, NY, 165–168, (2013), https://doi.org/10.1145/1453805.1453838
Ho, C., Mason, O., Spence, C.: An investigation into the temporal dimension of the mozart effect: Evidence from the attentional blink task. Acta Physiol. (Oxf) **125**(1), 117–128 (2007). https://doi.org/10.1016/j.actpsy.2006.07.006
Huang, R.H., Shih, Y.N.: Effects of background music on concentration of workers. Work **38**(4), 383–387 (2011). https://doi.org/10.3233/WOR-2011-1141
Ikeda, S., Oku, K., Kawagoe, K.: Music playlist recommendation using acoustic-feature transitions. In: Proceedings of the 9th International C* Conference on Computer Science & Software Engineering, ACM, New York, NY, 115–118, (2016), https://doi.org/10.1145/2948992.2949005

Johansson, R., Holmqvist, K., Mossberg, F., Lindgren, M.: Eye movements and reading comprehension while listening to preferred and non-preferred study music. Psychol. Music **40**(3), 339–356 (2011). https://doi.org/10.1177/0305735610387777

Kahneman, D.: Attention and Effort. Prentice-Hall, Englewood Cliffs, NJ (1973)

Kahneman, D.: Thinking. Fast and Slow, Farrar, Straus and Giroux, New York, NY (2011)

Kaminskas, M., Ricci, F.: Emotion-based matching of music to places. In: Tkalcic, M., Carolis, B.D, de Gemmis, M., Odic, A., Kosir, A. (eds) Emotions and Personality in Personalized Services, Springer, Cham, Switzerland, 287–310, (2017), https://doi.org/10.1007/978-3-319-31413-6_14

Knees, P., Schedl, M.: A survey of music similarity and recommendation from music context data. ACM Trans. Multimed. Comput. Commun. Appl. **10**(1), 2:1-2:21 (2013). https://doi.org/10.1145/2542205.2542206

Knees, P., Pohle, T., Schedl, M., Widmer, G.: A music search engine built upon audio-based and web-based similarity measures. In: Proceedings of the 30st International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, 447–454, (2007), https://doi.org/10.1145/1277741.1277818

Liang, D., Zhan, M., Ellis, D.P.W.: Content-aware collaborative music recommendation using pre-trained neural networks. In: Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR, Montreal, Canada, 295–301, (2015)

Liu, D., Bias, R.G., Lease, M., Kuipers, R.: Crowdsourcing for usability testing. In: Proceedings of the 75th ASIS&T Annual Meeting, ASIS&T, Silver Spring, MD, 1–10, (2012), https://doi.org/10.1002/meet.14504901100

Liu, H., Hu, J., Rauterberg, M.: iHeartrate: A heart rate controlled in-flight music recommendation system. In: Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research, ACM, New York, NY, 26:1–26:4, (2010), https://doi.org/10.1145/1931344.1931370

Logan, B., Chu, S.M.: Music summarization using key phrases. In: Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing, IEEE, New York, NY, 749–752, (2000), https://doi.org/10.1109/ICASSP.2000.859068

Lonsdale, A.J., North, A.C.: Why do we listen to music? a uses and gratifications analysis. Br. J. Psychol. **102**(1), 108–134 (2011). https://doi.org/10.1348/000712610X506831

Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G.: Recommender system application developments: a survey. Decis. Support Syst. **74**, 12–32 (2015). https://doi.org/10.1016/j.dss.2015.03.008

Lu, J., Zhao, P., Hoi, S.C.H.: Online passive-aggressive active learning. Mach. Learn. **103**(2), 141–183 (2016). https://doi.org/10.1007/s10994-016-5555-y

Mark, G., Gudith, D., Klocke, U.: The cost of interrupted work: more speed and stress. In: Proceedings of the 2008 ACM SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, 107–110, (2008), https://doi.org/10.1145/1357054.1357072

Mendes, C.G., Diniz, L.A., Miranda, D.M.: Does music listening affect attention? a literature review. Dev. Neuropsychol. **46**(3), 192–212 (2021). https://doi.org/10.1080/87565641.2021.1905816

Milliman, R.E.: The influence of background music on the behavior of restaurant patrons. J. Consumer Res. **13**(2), 286–289 (1986)

Moling, O., Baltrunas, L., Ricci, F.: Optimal radio channel recommendations with explicit and implicit feedback. In: Proceedings of the 6th ACM Conference on Recommender Systems, ACM, New York, NY, 75–82, (2012), https://doi.org/10.1145/2365952.2365971

Oliver, N., Flores-Mangas, F.: MPTrain: A mobile, music and physiology-based personal trainer. In: Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services, ACM, New York, NY, 21–28, (2006), https://doi.org/10.1145/1152215.1152221

van den Oord, A., Dieleman, S., Schrauwen, B.: Deep content-based music recommendation. In: Proceedings of the 27th Annual Conference on Neural Information Processing Systems, NeurIPS Foundation, San Diego, CA, 2643–2651, (2013)

Pampalk, E., Flexer, A., Widmer, G.: Improvements of audio-based music similarity and genre classificaton. In: Proceedings of the 6th International Conference on Music Information Retrieval, ISMIR, Montreal, Canada, 628–633, (2005a)

Pampalk, E., Pohle, T., Widmer, G.: Dynamic playlist generation based on skipping behavior. In: Proceedings of the 6th International Conference on Music Information Retrieval, ISMIR, Montreal, Canada, 634–637, (2005b)

Park, D.H., Kim, H.K., Choi, I.Y., Kim, J.K.: A literature review and classification of recommender systems research. Expert Syst. Appl. **39**(11), 10059–10072 (2012). https://doi.org/10.1016/j.eswa.2012.02.038

Proverbio, A.M., Nasi, V.L., Arcari, L.A., Benedetto, F.D., Guardamagna, M., Gazzola, M., Zani, A.: The effect of background music on episodic memory and autonomic responses: listening to emotionally touching music enhances facial memory capacity. Scientific Rep. (2015). https://doi.org/10.1038/srep15219

Rieck, K., Trinius, P., Willems, C., Holz, T.: Automatic analysis of malware behavior using machine learning. J. Comput. Secur. **19**(4), 639–668 (2011). https://doi.org/10.3233/JCS-2010-0410

Silla, C.N., Jr., Freitas, A.A.: A survey of hierarchical classification across different application domains. Data Min. Knowl. Disc. **22**(1–2), 31–72 (2011). https://doi.org/10.1007/s10618-010-0175-9

Song, Y., Dixon, S., Pearce, M.: A survey of music recommendation systems and future perspectives. In: Proceedings of the the 9th International Symposium on Computer Music Modeling and Retrieval, Springer, Berlin, Heidelberg, 395–410, (2012)

Takahashi, R., Ohishi, Y., Kitaoka, N., Takeda, K.: Building and combining document and music spaces for music query-by-webpage system. In: Proceedings of the 9th Annual Conference of the International Speech Communication Association, ISCA, Baixas, France, 2020–2023, (2008)

Tanaka, T., Fujita, K.: Study of user interruptibility estimation based on focused application switching. In: Proceedings of the 2011 ACM Conference on Computer Supported Cooperative Work, ACM, New York, NY, 721–724, (2011), https://doi.org/10.1145/1958824.1958954

Tateyama, Y., Matsumoto, Y., Kagami, S.: Concentration detection by eye movements: Towards supporting a human. In: Proceedings of the 2004 IEEE International Conference on Systems, Man & Cybernetics, IEEE, New York, NY, 1544–1548, (2004), https://doi.org/10.1109/ICSMC.2004.1399851

Tzanetakis, G., Cook, P.: MARSYAS: a framework for audio analysis. Organised Sound **4**, 169–175 (2000). https://doi.org/10.1017/S1355771800003071

Volokhin, S., Agichtein, E.: Towards intent-aware contextual music recommendation: initial experiments. In: Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, 1045–1048, (2018), https://doi.org/10.1145/3209978.3210154

Wang, Y.: A hybrid recommendation for music based on reinforcement learning. In: Proceedings of the 24th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, Cham, Switzerland, 91–103, (2020), https://doi.org/10.1007/978-3-030-47426-3_8

Wells, A.: Popular music: emotional use and management. J. Pop. Cult. **24**(1), 105–117 (1990). https://doi.org/10.1111/j.0022-3840.1990.00105.x

Yakura, H., Nakano, T., Goto, M.: Focusmusicrecommender: A system for recommending music to listen to while working. In: Proceedings of the 23rd ACM International Conference on Intelligent User Interfaces, ACM, New York, NY, 7–17, (2018), https://doi.org/10.1145/3172944.3172981

Zhang, S., Yao, L., Sun, A., Tay, Y.: Deep learning based recommender system: A survey and new perspectives. ACM Comput. Surv. **52**(1), 5:1-5:38 (2019). https://doi.org/10.1145/3285029

Züger, M., Fritz, T.: Interruptibility of software developers and its prediction using psycho-physiological sensors. In: Proceedings of the 2015 ACM SIGCHI Conference on Human Factors in Computing Systems, ACM, New York, NY, 2981–2990, (2015), https://doi.org/10.1145/2702123.2702593

Hiromu Yakura received the B.E. degree in 2019 and the M.E. degree in 2021, both from University of Tsukuba, Japan. He is currently pursuing a Ph.D. degree at University of Tsukuba. His research interests lie in the intersection of machine learning and human-computer interaction, which involves various application areas such as creativity support, human resource development, and virtual reality. He is a recipient of Google Ph.D. Fellowship and Microsoft Research Asia Fellowship.

Tomoyasu Nakano received the Ph.D. degree in Informatics from University of Tsukuba, Tsukuba, Japan in 2008. He is currently working as a Senior Researcher at the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. His research interests include singing information processing, human-computer interaction, and music information retrieval. He has received several awards including the IPSJ Yamashita SIG Research Award from the Information Processing Society of Japan (IPSJ), the Best Paper Award from the Sound and Music Computing Conference 2013, and the Honor-

able Mention Poster Award from the IEEE Pacific Visualization Symposium 2018. He is a member of the IPSJ and the Acoustical Society of Japan.

Masataka Goto received the Doctor of Engineering degree from Waseda University in 1998. He is currently a Prime Senior Researcher at the National Institute of Advanced Industrial Science and Technology (AIST), Japan. Over the past 30 years he has published more than 300 papers in refereed journals and international conferences and has received 57 awards, including several best paper awards, best presentation awards, the Tenth Japan Academy Medal, and the Tenth JSPS PRIZE. He has served as a committee member of over 120 scientific societies and conferences, including the General Chair of ISMIR 2009 and 2014. As the research director, he began OngaACCEL project in 2016 and RecMus project in 2021, which are five-year JST-funded research projects (ACCEL and CREST) related to music technologies.