FocusMusicRecommender: A System for Recommending Music to Listen to While Working

Hiromu Yakura
University of Tsukuba
Tsukuba, Japan
hiromu@coins.tsukuba.ac.jp

Tomoyasu Nakano **Masataka Goto**
National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan
{t.nakano, m.goto}@aist.go.jp

ABSTRACT

This paper proposes *FocusMusicRecommender*, an automated system recommending background music to listen to while working. Recommendation systems matching user preferences have been widely researched even though research has shown that music that listeners strongly like is not suitable background music because it interferes with their concentration. FocusMusicRecommender plays songs that users may "neither like nor dislike" instead of "like very much." It is designed to by default summarize a song automatically so that users can give "like very much" feedback by pressing a "keep listening" button or "dislike very much" feedback by pressing a "skip" button. It uses this feedback, along with users' concentration levels estimated from their behavior history, to distinguish between the preference levels "like" and "like very much." It then estimates the preference levels of unplayed songs and selects the most suitable song by considering the user's current concentration level. The effectiveness of the proposed feedback method and suitability of the recommendation results were verified experimentally and in user studies. Furthermore, it is confirmed that the proposed method can estimate the user's concentration level more accurately than the previous methods.

CCS Concepts

•Human-centered computing → HCI design and evaluation methods; •Applied computing → Sound and music computing;

Author Keywords

Music recommendation; concentration level estimation; background music.

INTRODUCTION

Many people listen to music while working. In the survey of 189 students by Lonsdale et al. [21], 75.7% of the participants confirmed that they listen to music when working or studying, and most of them reported that they listened because it helped them concentrate. The efficacy of music as a study aid is also suggested by Figure 2 taken from Google Trends[1], which presents the popularity of search queries in YouTube[2]. It shows that the term "study music" is used as frequently as "classical music" which is often used in psychological studies to measure the effect of background music [25, 12, 27, 15]. We therefore propose *FocusMusicRecommender*, a system recommending background music suitable for listening to while working and thereby improve the efficiency and the quality of one's work.

IUI'18, March 07–11, 2018, Tokyo, Japan

ISBN 978-1-4503-4945-1/18/03. . . $15.00

DOI: https://doi.org/10.1145/3172944.3172981

Figure 1. Usage of FocusMusicRecommender. It selects songs automatically according to the user's feedback and behavior history and plays in an abridged manner.

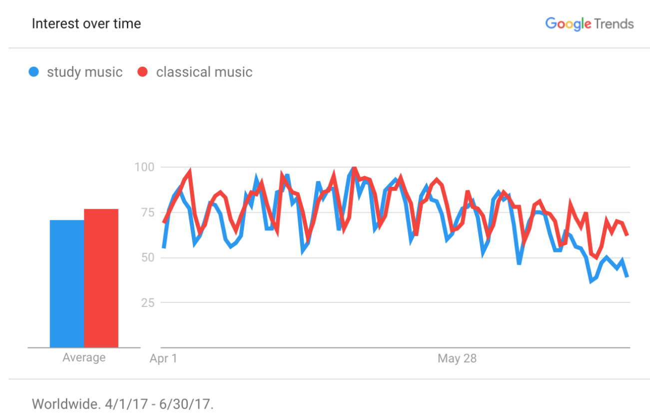

Figure 2. Comparison of the interest of the terms "study music" and "classical music" in YouTube[3]. The search interest of "study music" is as high as that of "classical music," and they drop on weekends.

The conventional systems for recommending a song that the user would like the most [28] are unsuitable because a song

[1] https://trends.google.com/
[2] https://www.youtube.com/
[3] This result can be inferred from https://g.co/trends/4Zhu7.

strongly preferred by the user may interfere with the user's concentration. This is supported by a survey by Huang et al. [15] on the relation between concentration level while listening to songs and the preference level for the songs rated using a five-point Likert scale of "like very much," "like," "neither like nor dislike," "dislike," and "dislike very much." They concluded that the concentration level of people listening to songs they liked or disliked very much was much worse than it was in a silent environment whereas that of people listening to songs they liked, neither liked nor disliked, or disliked was not significantly different from what it was in a silent environment. They also concluded that the concentration level was affected more by the user's preference level than by the genre of the background music. In other words, to concentrate on work it is more important to avoid songs that arouse strong emotions than to select appropriate music genres.

The user can, of course, take this point into consideration when selecting songs, but considering how much one likes each song while working is troublesome. As shown in Figure 1, FocusMusicRecommender therefore selects songs the user may neither like nor dislike automatically according to the user's working behavior (mouse operation, etc.) and the user's listening feedback elicited by the songs (skipping, etc.) and plays them continuously.

FocusMusicRecommender also plays songs in an abridged manner by using chorus section information. The song ends after the first chorus section. This was inspired by the fact that many video clips posted on video-sharing services concatenate chorus sections of multiple songs for the purpose of listening while working[4]. It lets a user encounter unknown music efficiently by listening to various songs and does this without burdening the user. In detail, we introduce a "keep listening" button to make it possible for the user to give feedback such as "I want to listen more because it is my favorite song" as well as a "skip" button for giving feedback such as "I want to skip this song because I dislike it" [24].

Furthermore, FocusMusicRecommender estimates the user's concentration level and takes it into account for recommendation. Based on the hypothesis that feedback given when concentrating reflects the preference level for songs better than does feedback given when not concentrating, the system combines the estimated concentration level with the feedback to determine the preference level precisely. In addition, when there are multiple songs judged suitable to be recommended, the system selects one by adjusting the selection criterion according to the concentration level.

RELATED WORK

In this section we describe related research on music recommendation for specific situations, music recommendation based on limited feedback, and estimation of concentration level during working with a computer keyboard and mouse. The situation of working with a keyboard and mouse is not limited to desktop computers and laptops and may be similar to that of working with a tablet or the like, so hereinafter these kinds of devices are collectively referred to as personal computers.

Music Recommendation for Specific Situations

Many methods of music recommendation for specific situations have been proposed [23, 22, 18, 1]. For example, PAPA proposed by Oliver et al. [23] uses the user's heart rate as feedback in order to assist physical exercise. The system plays faster songs (more beats per minute) when the heart rate is low and plays slower songs when the heart rate is high. InCarMusic proposed by Baltrunas et al. [1] is targeted to support driving of a car and changes the selection priority of music genres according to whether the user is traveling an ordinary road or an expressway, whether the user is sleepy or alert, and so on.

Although music recommendation systems using recent machine learning algorithms have been proposed [19, 32], none of them recommend background music suitable for listening to while working because they don't prioritize songs that the user neither likes nor dislikes. That is one of the reasons why Demetriou et al. [8] recommended that the music information retrieval community develop recommendation systems optimizing the user's level of cognitive engagement rather than to meeting the user's preference.

Music Recommendation on Limited Feedback

To design a music recommendation system to be used while working, suppressing the burden of giving feedback is important. For example, Pampalk et al. [24] proposed a method that recommends music on extremely limited feedback from users. It uses as feedback only the skip operation that the user either performed or did not perform while a song was playing: songs that were not skipped are regarded as liked and songs that were skipped are regarded as disliked.

1. For each candidate song, let s_s be the musical similarity to the nearest song skipped and let s_a be the similarity to the nearest song that was not skipped.
2. If there are candidate songs that satisfy $s_a > s_s$, play the one with the largest s_a.
3. Otherwise, play the song with the largest $\frac{s_a}{s_s}$.

This method uses only the two preference levels determined according to whether or not the user skips a song – i.e., "dislike very much" or "not dislike very much" – and thus cannot give priority to songs that are neither liked nor disliked when recommending music to listen to while working.

Concentration Level Estimation

Since FocusMusicRecommender recommends in accordance with the user's concentration level, it should estimate the concentration level automatically. Estimating "concentration" or "interruptibility" level during work on a personal computer has been investigated in several studies [30, 9, 35, 29]. For example, Fogarty et al. [9] used not only the numbers of keyboard and mouse operations but also the door-opening count

[4]Searching "*Sagyouyou BGM* (in Japanese)," which means background music to be listened to while working, in YouTube gives clips that have titles like "*Chorus medley* (in Japanese)" and concatenate summarized version of multiple songs.

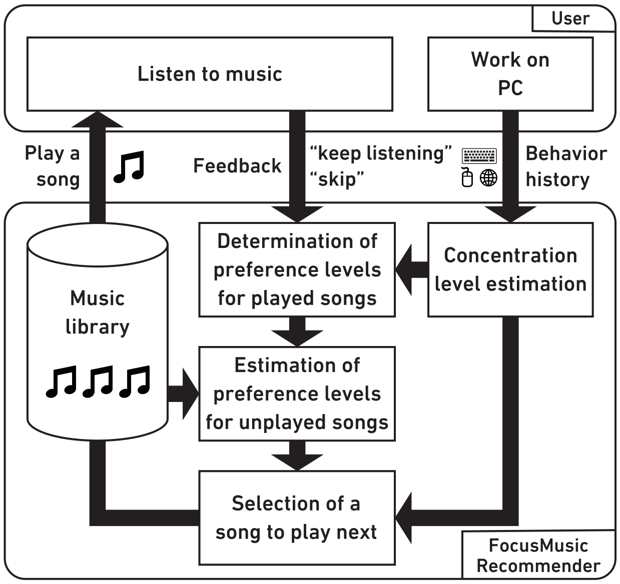

Figure 3. Overview of FocusMusicRecommender. It determines the user's preference levels for songs and selects the song played next according to the user's feedback and behavior history.

detected by a magnetic sensor. Züger et al. [35] used a biometric sensor to measure the user's electroencephalogram, skin potential, and heart rate. Tanaka et al. [29] proposed a method combining the numbers of keyboard and mouse operations and the number of switchings of active applications.

Using physical sensors or biosensors for music recommendation, however, is considered unrealistic in terms of costs and the psychological barriers of users. The method proposed by Tanaka et al. [29] uses features that can be collected without using external sensors, but it uses the number of specific operations in a fixed length of time and therefore cannot be applied during music playback where the length of time is variable because a song may be skipped after only a few seconds.

FOCUSMUSICRECOMMENDER

In this section we describe the overview of FocusMusicRecommender (Figure 3) and describe the methods that determine the user's preference level for played songs and select the songs to be played next.

Overview

FocusMusicRecommender is a system that helps users concentrate by listening to music when working on a personal computer. Although a user could use an automatic playback function in order to avoid the trouble of bothering to select songs during work, random playback or the conventional recommendation method plays songs the user may like very much, which interferes with the user's concentration. The proposed system therefore automatically selects songs the user neither likes nor dislikes and summarizes them by successively playing each from its beginning to the end of the first chorus section.

Many methods of music summarization using chorus section information have been proposed [20, 5, 7]. Along with the user-generated summarized clips described in Section *"Introduction"*, the acceptance of such automated summarization function in music players is supported by the fact that similar functions are implemented in the Walkman portable music player[5] and the continuous playback function[6] of the music browsing assistance service Songrium [13], and therefore it is considered to be effective. Furthermore, the function enables the system to introduce a "keep listening" button, which is necessary to avoid playing songs the user may like very much. Without the function, the system would need to require the user to explicitly input the preference level song by song, which would distract the user from his/her work.

The system therefore determines the preference level according to both user feedback such as "keep listening" or "skip" and the concentration level estimated from the user's behavior history on the personal computer. Furthermore, to choose the most suitable of multiple candidate songs it adjusts the selection criterion according to the estimated concentration level.

Determination of User's Preference Level for Played Songs

In this section we describe in detail a method for determining a user's preference level for played songs. As shown in Figure 4, we first introduce a "keep listening" button to determine the three levels "like very much," "neither like nor dislike," and "dislike very much" and then extend the determination to five levels by using the user's concentration level.

User Feedback from a "keep listening" Button

Getting user feedback from a "keep listening" button extends the method of Pampalk et al. [24], described in Section *"Music Recommendation on Limited Feedback"*, and enables the system to eliminate not only songs the user dislikes very much but also songs the user likes very much. While songs are being summarized the system determines their preference levels according to the following user feedback.

- **Do nothing**
  It plays the song until the end of the first chorus section and judges that the user neither likes nor dislikes the song. Each song is played back at least 30 seconds because it is difficult for the user to decide right after the playback starts whether to skip a song or keep listening to it.

- **Press "skip" button**
  It judges that the user dislikes the song very much, and it plays the next song immediately.

- **Press "keep listening" button**
  It judges that the user likes the song very much, and it plays it until the end.

[5] ZAPPIN Playback: https://docs.sony.com/release/NWZW273S_W274S_guide_EN.pdf
[6] Songrium Smart Player: http://smart.songrium.jp/

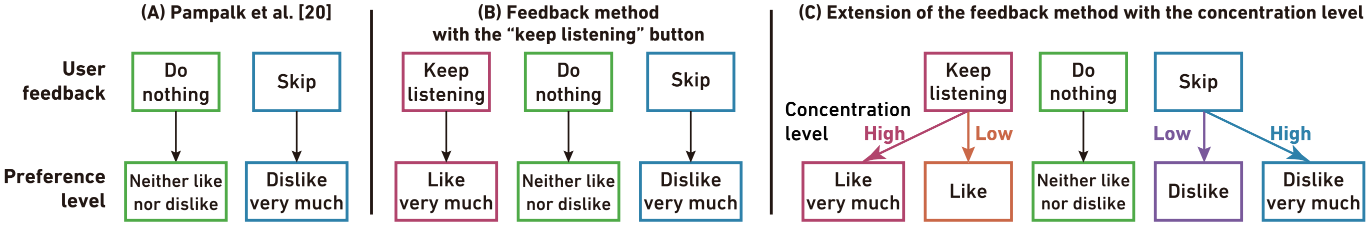

Figure 4. Relations between user feedback and preference levels. (A) From the "skip" feedback, only "dislike very much" or not can be obtained. (B) Adding "keep listening" feedback makes three preference levels, including "neither like nor dislike," available. (C) Five levels, including "like" and "dislike," can be obtained by combining the concentration level.

Table 1. Preference level can be obtained on a five-point scale by combining user feedback and concentration level as indicated in this table and shown in Figure 4(C).

Concentration	User Feedback: Keep listening	Do nothing	Skip
High	Like very much	Neither like nor dislike	Dislike very much
Low	Like	Neither like nor dislike	Dislike

Refine Preference Level Using Estimated Concentration Level

After obtaining the preference in "like very much," "neither like nor dislike," and "dislike very much" levels by the method described in Section *"User Feedback from a "keep listening" Button"*, the system determines it in five levels including "like" and "dislike" by using the user's concentration level and 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.

The system determines the preference level by combining the user feedback and the concentration level as summarized in Table 1.

The reason we extend the feedback by using the concentration level is that discriminating liked songs from liked very much songs is more suitable for the problem situation. In detail, according to [15], misclassifying liked songs as neither liked nor disliked may not affect the concentration of the user as much as not discriminating between songs that are liked and songs that are liked very much would. We therefore enable the system to deal with such prioritization by combining the five-level preference with hierarchical classification algorithms [17] as shown in Figure 5. This helps the system to reduce the possibility of misclassifying songs the user likes or dislikes very much as songs the user neither likes nor dislikes, which would interfere with the user's concentration.

Song Selection Method Corresponds to Concentration Level

The system estimates the preference levels of songs that have not been played yet relative to those of previously played songs and plays songs the user may neither like nor dislike. If

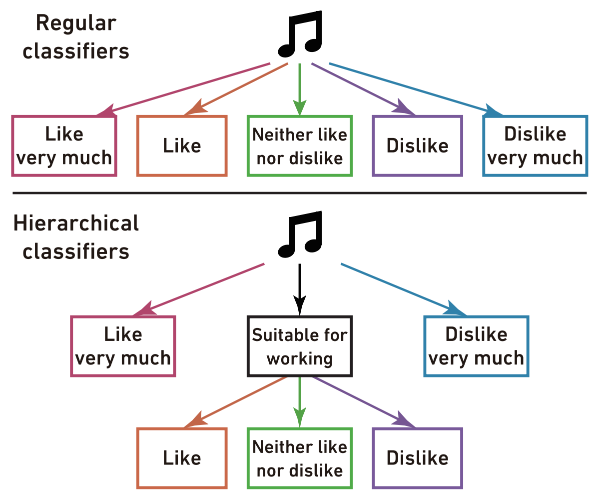

Figure 5. Difference between regular classifiers and hierarchical classifiers in FocusMusicRecommender. The extension of the feedback to five levels enables the hierarchical classifier to treat liked songs, neither liked nor disliked songs, and disliked songs as belonging to a subgroup of songs suitable to be listened to while working.

there are two or more songs estimated to be neither liked nor disliked, the system needs to choose one of them.

For the purpose of enhancing the user's concentration, it adjusts the selection criterion according to the current level of concentration. When the concentration level is high, priority is given to music similar to the song that was played immediately before in order to avoid sudden changes that might distract the user. On the other hand, when it is low, priority is adjusted to play a variety of songs in order to give the user a chance to change his/her mood. These priority adjustments are based on the results of a survey by Wells [34]: 73.3% of the 225 participants agreed that music is useful for changing their mood and that the genres of music that they listen to for changing their mood are very diverse.

IMPLEMENTATION

As shown in Figure 6, FocusMusicRecommender consists of four modules: playback, behavior history collection, concentration level estimation, and music selection. In this section we describe the implementation of each module.

Playback

The playback module plays the selected song in cooperation with the other three modules by using Songle Widget [11], which is an open framework for providing an embeddable

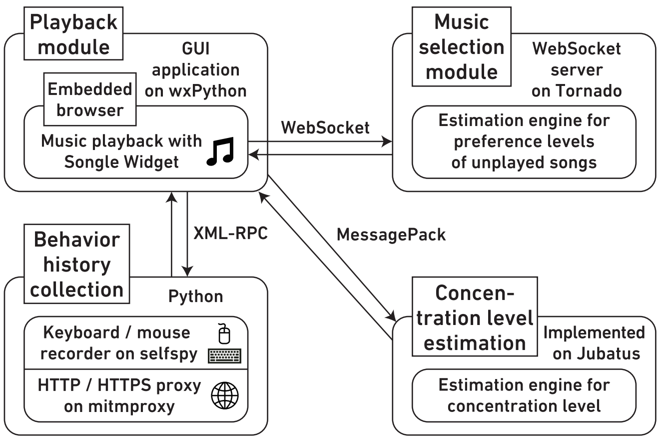

Figure 6. Structure of FocusMusicRecommender. The playback module handles the interface, and the other three modules operate in the background.

music player. Songle Widget makes playback easy by using chorus section information that is estimated automatically and can be corrected manually on Songle [10].

Behavior History Collection

The behavior history collection module records three types of action logs: keyboard input, mouse input, and Web communication (Table 2). The reason we record Web communication, which is not used in existing related methods [30, 9, 35, 29], is that it reflects the work content, such as whether the user is performing a web search or using a social networking service.

Concentration Level Estimation

The concentration level estimation module uses the online learning algorithm AROW [6] with the n-gram of the hash value of the behavior history. This method not only solves the problem described in Section *"Concentration Level Estimation"* but also can use more information such as application name and host name, than would be available if only the number of operations were used. For example, Google and Facebook accesses have different meanings in this method because they produce different hash values, while the counting method treats Google and Facebook accesses equally. This method was inspired by malware detection methods that use n-grams of the hash values of the system call for learning [26, 2].

There are several reasons we use AROW for the concentration level estimation [6]:

- It is suitable when collecting a large amount of labeled data is difficult because it uses passive-aggressive update algorithm and therefore converges faster than other algorithms.
- It is suitable when the feature space is sparse because it uses a confidence-weighted algorithm to take into account the frequency of features.
- It is suitable when the labeled data may contain noise such as the subjective deviance of the concentration level because it uses a mistake bound that doesn't assume separability.

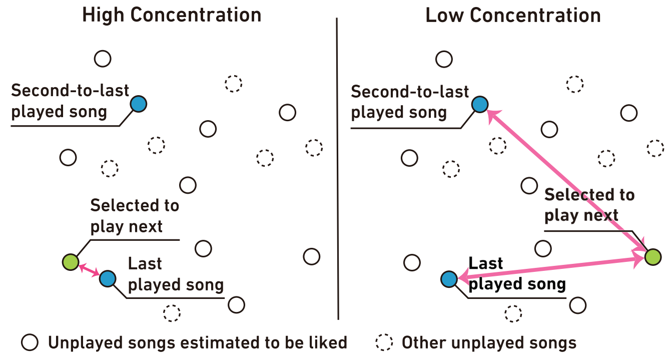

Figure 7. Overview of the song selection method. When the concentration level is high, the song most similar to the last played song is selected. When it is low, the song most dissimilar to the two previous songs is selected.

Music Selection

As mentioned in Section *"Song Selection Method Corresponds to Concentration Level"*, the music selection module estimates the preference levels for unplayed songs relative to those for played songs. This module uses HierCost [4] based on user's preference-level information obtained by the method described in Section *"Determination of User's Preference Level for Played Songs"*. The reasons for using HierCost are that it is one of the hierarchical classification algorithms that can be used for educational and research purpose and also that it is designed to deal with unbalanced data. Even though the number played songs is limited, HierCost's ability to deal with unbalanced data reduces the chance of playing songs that interfere with a user's concentration because they are songs the user likes or dislikes very much but were misclassified as songs the user neither likes nor dislikes.

When selecting the song to be played next, the system changes the criterion in accordance with the concentration level estimated as described in Section *"Song Selection Method Corresponds to Concentration Level"*. Although several recommendation methods [3, 16] consider songs recently played, our proposed method also considers both the estimated preference levels and the current concentration level as follows (Figure 7):

1. To start making a listening history, select the first and second songs randomly.
2. Estimate the preference levels of unplayed songs and list as candidates all songs labeled as "neither like nor dislike." If there are none, select candidates in the order of songs labeled as "like," "dislike," "like very much," and "dislike very much," according to the effect on the concentration level described in [15].
3. Select the third (next) songs from the candidates based on the musical similarity like [24] as follows.
   (a) When the user's concentration level is high, in order to avoid sudden changes that might distract the user, select the song that is the most similar to the last played song (the song with the maximum similarity to the song played immediately before).

Table 2. List of user behaviors collected by FocusMusicRecommender.

Types of Behavior	Collected Information	Examples
Keyboard input	"key [Target application name] [Key]" (Modifier keys are put together in a single event.)	"key Google Chrome a" (Press "a" in Google Chrome)
		"key Microsoft Excel <[Ctrl: v]>" (Press "Ctrl+v" in Microsoft Excel)
Mouse input	"mouse [Target application name] [Event number]" (Event numbers 1 $\sim$ 7 respectively represent left, middle, and right button clicks and up, down, left, and right scrolling.)	"mouse Firefox 1" (Click a left button in Firefox)
		"mouse Skype 4" (Scroll up in Skype)
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 www.facebook.com" (Send a POST request to www.facebook.com)

(b) When the user's concentration level is low, in order to give the user a chance to change his/her mood [34], select the song that is the least similar to both the last played song and the second-to-last played song (a song with the minimum sum of similarities to the two songs played immediately before). This selection is based on the two previous songs. Otherwise, the problem that songs in two genres are selected alternately would occur, reducing the diversity of the songs played.

4. Go back to 2.

EVALUATION

We first evaluated the accuracy of the concentration level estimation and validity of the preference level determination experimentally and then conducted user studies. In this section we describe detailed procedures and their results.

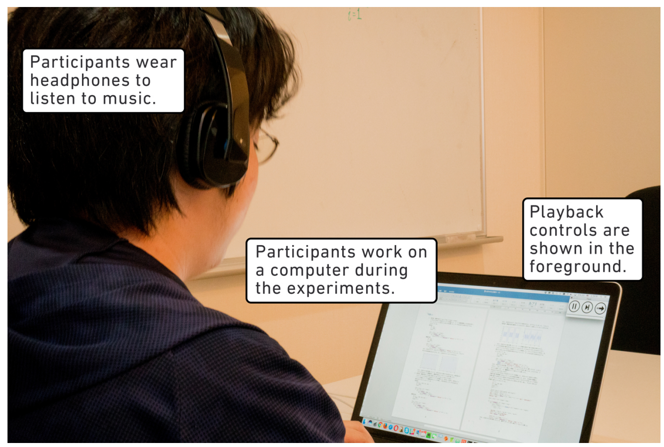

Figure 8. Example of experimental setup. Participants listened to music using earphones or headphones while working on a personal computer in a quiet room.

Data

We used top 50 most frequently played songs with the tag "VOCALOID," which are songs created using a popular singing synthesis software, in the popular Japanese video-sharing service *NicoNico Douga* (http://nicovideo.jp/). Our experiments used VOCALOID songs because accurate user-corrected chorus sections for all 50 songs are available via Songle Widget. Furthermore, those songs tend to cover diverse genres[7] because they are created in a user-generated content community [14, 13]. In fact, 15 different tags indicating the genres are used for the songs such as "*VOCAROCK* (tags used for rock music)," "*Vocaloid Japanese-style music*," and "*Mikuno-Pop* (tags used for electro-pop music)."

The similarity between songs was estimated based on three-dimensional musical feature vector calculated in Songrium [13]. First, MARSYAS [31] was used to obtain a 35-dimensional feature vector for more than 98,000 VOCALOID songs. 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 (PCA) to the feature vector for dimensionality reduction.

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

Experiments

To confirm the effectiveness of the proposed methods, we conducted two experiments using data obtained from users. The first evaluated the concentration level estimation described in Section *"Implementation"* and the second evaluated the preference level determination described in Section *"Determination of User's Preference Level for Played Songs"*.

Collection of User Data for Evaluation

These experiments involved eight voluntary participants (male students 17 to 24 years old) who were in the habit of

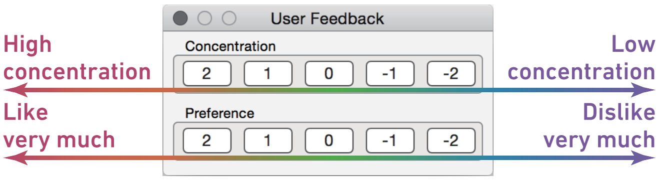

Figure 9. Dialog box used for entering the concentration level and preference level as validation data. The system asks users to enter these data each time a song is played.

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

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

listening to music while working on a personal computer. In a quiet room they listened to all 50 target songs in random order using headphones while being presented with "keep listening" and "skip" buttons. During the experiments, four participants wrote new documents by using word processing applications and the other four worked on programming on a personal computer (Figure 8). Their behavior histories were collected automatically, and they were informed beforehand that the history data is saved in the form of irreversible hashes so that they can work as they usually do.

Their concentration levels and preference levels were also obtained using a dialog box (Figure 9) presented each time a song was played. The method of obtaining validation data by dialog box was used in previous studies of concentration level estimation [9, 29], and as in those studies we used 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.

Accuracy of Concentration Level Estimation

Verifying by five-fold cross-validation the correspondence of the estimated concentration level and the validation data that the participants entered, we found that the accuracy was 37.5% in five-class estimation and 70.5% in two-class estimation. Table 3 and Table 4 present the respective confusion matrices.

Although the accuracy of the two-class estimation was lower than that of the method using the physical sensor (76.9%) [9]

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

Estimated Label	User-entered Label High $(2, 1)$	Low $(0 \sim -2)$
High concentration	99	64
Low concentration	54	183
Total	153	247

and that of the method using the biometric sensor (78.6%) [35], it was higher than that of the method using features that can be collected without external sensors (58.4%) [29]. Therefore the effectiveness of the proposed method for estimating the concentration level was confirmed.

We compared the accuracy of the concentration levels estimated from behavior history with and without consideration of Web communication history. The result is shown in Table 5, and it indicates that the Web communication history improves the accuracy of the estimated concentration levels.

In calculating the above accuracies of the previous methods [9, 29] and the proposed method, we converted the multiclass estimation results into the two-class estimation results by following the procedure of Züger et al. [35]: a concentration level labeled "neither high nor low" (= 0) is categorized as "low."

Validity of Preference Level Determination

First we evaluated the validity of the "keep listening" feedback describe in Section *"User Feedback from a "keep listening" Button"* by checking 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 6, and Pearson's correlation coefficient between the preference level the participants entered and the user feedback is 0.55 $(p < 0.01)$. Note that p-value here indicates the probability that the data would have arisen if there is no correlation.

Then we evaluated whether the determination method that combines the concentration level estimation with the "keep listening" feedback appropriately reflects the user's preference level. The result is shown in Table 7, and Pearson's correlation coefficient between the preference level determined by the proposed method and the one that the participants entered is 0.67 $(p < 0.01)$. Therefore the hypothesis mentioned in Section *"Refine Preference Level Using Estimated Concentration Level"* was supported and the appropriateness of the proposed determination method shown in Table 1 was confirmed.

Moreover, in Table 7, we can find quite a few "like" and "dislike" songs in the "do nothing" row. It suggests that users at the high concentration level don't pay attention to songs that they neither "like very much" nor "dislike very much" as shown in Figure 4(C). In other words, it confirms that songs disliked, liked, or neither liked nor disliked don't interfere with users' concentration as stated in [15].

User Study

In the user study, the eight participants in the experiments described in Section *"Collection of User Data for Evaluation"* used FocusMusicRecommender and the comparison implementations described below.

Skipping Behavior (SB)

We implemented as a baseline for comparison a system based on the conventional method proposed by [24] that uses the skip operation as feedback. Instead of avoiding songs similar to songs the user skipped, the system

Table 5. Comparison of the accuracy of the concentration levels estimated with and without using Web communication history. The accuracy obtained using Web communication history was higher than that obtained not using Web communication history.

	Without Web Communication (only mouse and keyboard)	With Web Communication
5-class	32.4%	37.5%
2-class	61.9%	70.5%

Table 6. Correspondence between the user-entered preference level and the "keep listening" and "skip" feedback ($r = 0.55, p < 0.01$).

User Feedback	Preference Level: Like very much (2)	Like, Neither like nor dislike, Dislike ($1 \sim -1$)	Dislike very much (-2)	Total
Keep listening	43	64	2	109
Do nothing	13	179	6	198
Skip	1	52	40	93
Total	57	295	48	400

Table 7. Correspondence between the user-entered preference level and the combination of the estimated concentration level and the user feedback ($r = 0.67, p < 0.01$).

User Feedback (Concentration Level)	Preference Level: Like 2	1	0	Dislike -1	-2	Total
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

Table 8. Distribution of the preference levels and the number of operations for the songs played by SB, FMR-1, and FMR-2. The proposed method plays fewer songs that the participants like very much or dislike very much and would decrease the concentration level.

	Preference Level: Like 2	1	0	Dislike -1	-2	Participants pressed "skip" or "keep listening"
SB	43	62	79	38	18	102
FMR-1	30	45	111	34	20	91
FMR-2	22	48	121	38	11	70

plays songs that are dissimilar to songs for which the user pressed either a "skip" or "keep listening" button. In other words, we extended the binary feedback of [24] shown in Figure 4(A) to the three levels shown in Figure 4(B) in order to avoid songs the user may like or dislike very much.

FocusMusicRecommender Not Considering Concentration Level (FMR-1)

We implemented an alternative version of FocusMusicRecommender that doesn't take the user's concentration level into account. The system estimates the preference level of unplayed songs in the three levels using the feedback shown in Figure 4(B), and thus, the estimation process is excepted to perform like regular classifiers that don't use the hierarchical information. Then instead of changing the selection criterion as described in Section *"Music Selection"*, the system selects the song to be played next randomly from unplayed songs estimated to be neither liked nor disliked.

FocusMusicRecommender with Concentration Level (FMR-2)

We of course also implemented the proposed version of FocusMusicRecommender. The system uses the five-level preference based on the estimated concentration level shown in Figure 4(C). The system also changes the criterion for selecting songs to be played next in accordance with the concentration level as described in Section *"Music Selection"*.

The participants used each implementation until 30 of the 50 songs had been played under the same condition of work contents and environments, and they commented on the ease of use and their impressions of the songs played. The data collection described in Section *"Collection of User Data for Evaluation"* was carried out at least one month before the user study in order to avoid anomalous results due to short-term preference changes caused by repeated listening.

Result

Table 8 shows the distribution of the preference level the participants entered beforehand in Section *"Collection of User Data for Evaluation"* and the number of operations the participants performed for songs played by SB, FMR-1, and FMR-2. In contrast to the distribution of the population shown in the bottom row of Table 7, the hypergeometric p-value of avoiding "like very much" or "dislike very much" songs in SB, FMR-1, and FMR-2 is 0.1×10^{-1}, 2.4×10^{-7}, and 1.6×10^{-13}, respectively. The results demonstrate that FMR-

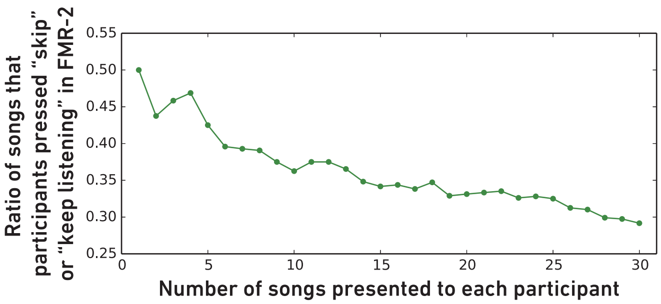

Figure 10. Transition of the ratio of songs that participants pressed the "skip" or "keep listening" button while listening to the recommendation results of FMR-2.

2, which implements the proposed method, played fewer songs that would decrease the concentration level than SB and FMR-1 did and thus 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 SB and FMR-1 did.

The difference between the results of SB and FMR-1 is due to FMR-1 deciding what song to play by using a learning algorithm to estimate the preference levels of unplayed songs and SB basing the decision on the neighbors of each song. The difference between FMR-1 and FMR-2 is due to the precise determination of the preference level as described in Section *"Refine Preference Level Using Estimated Concentration Level"*. 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, as shown in Table 8, FMR-2 played fewer "like very much" and "dislike very much" songs than FMR-1 even though it played more "like" and "dislike" songs, resulting in fewer interruptions.

In addition, it is suggested the proposed method is resistant to the *cold-start* (*new-user*) problem [33], which is that recommendation systems cannot handle new users. Figure 10 shows that the ratio of the numbers of songs for which participants pressed the "skip" or "keep listening" button dropped as the number of played songs increased and stabilized after about ten songs. This is because HierCost, the classification algorithm used in FMR-2, bases its calculation of misclassification cost on both the hierarchical information and the population of labeled data in order to deal with unbalanced data.

User Comments

The participants commented positively about the songs automatically played by the proposed system, saying things like "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." Regarding the "keep listening" feedback, comments such as "I didn't feel it burdened me," "I didn't particularly mind it," and "I didn't feel uncomfortable" were obtained. The participants also commented positively about the "automatic summarization" function, saying things like "this function is convenient because I can listen to many songs without being bored."

One participant mentioned that using the "keep listening" button seemed to interrupt work more than using the "skip" button did. It is presumed that this is attributable to the fact that a "skip" button is widely used by many music players and a "keep listening" button is not. The participant 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 confirmed. Moreover, because an increase in the number of played songs leads to an improvement in the accuracy of estimating the preference level of unplayed songs, the more the user uses the proposed system, the less often it would recommend songs that are not suitable for listening while working. As shown in Figure 10, if the user continued to use it, we would expect the number of button operations to decrease, resulting in fewer interruptions.

CONCLUSION AND FUTURE WORK

We have described FocusMusicRecommender, a music recommender system designed to improve users' concentration while working on personal computers. With a focus on background music for enhancing concentration, we designed the system to give priority to songs that a user may neither like nor dislike rather than like very much. The system's design is also consistent with the fact that the effects on a user's concentration level of liked, neither liked nor disliked, and disliked songs are not significantly different, while songs the user liked very much interfere with his/her concentration.

We introduced a feedback method that obtains the three levels of preference "like very much," "neither like nor dislike," and "dislike very much" while suppressing the burden on the user by implementing a "keep listening" button in addition to a "skip" button. Furthermore, we enabled the preference level to be acquired more precisely by determining the degree of "like" or "dislike" according to the user's automatically estimated concentration level. Hypothesizing that feedback given when concentrating reflects the preference level for songs better than does feedback given when not concentrating, we proposed a method for estimating the preference levels of unplayed songs and selecting the most suitable song by taking into account the relationship between the concentration level and the preference levels. The results of evaluation experiments supported our hypothesis and showed that the proposed system has high recommendation performance and is suitable for use during work.

We also proposed a method for estimating the concentration level from the user's behavior history collected without using physical sensors and confirmed it can estimate the user's concentration level more accurately than the previous methods used in experiments like those described in Section *"Concentration Level Estimation"*. This method can be applied not only for music recommendation but also for avoiding unwanted interruptions while working like the conventional methods.

Future Work

For future work, we would like to expand the experiments with alternative approaches along with increasing the number of the participants and the songs. For example, we are considering a different music selection method that takes into account novelty and diversity of the recommended songs because they are sometimes considered in the quality assessment of recommendation systems. Moreover, since the proposed system plays songs in an abridged manner, we would like to use other musical feature extraction methods that take into consideration the structure and variation in the song.

Furthermore, we would like to explore the new interactions due to using the estimated concentration level. For example, FocusMusicRecommender can prompt the user to take a break when the estimated concentration level stays low. Since it is possible to play songs the user may like very much, as the conventional recommendation systems do, by changing the priority of recommendation described in Section *"Music Selection"*, the system can help the user change the his/her mood seamlessly when he/she accepts the recommendation of taking a break. In addition, whether the user did or did not accept the recommendation is the information that can be used as labeled data for the estimation. In other words, since the system uses an online learning algorithm to estimate the concentration level, it is expected to be personalized and made more accurate by collecting such data.

ACKNOWLEDGMENTS

This work was supported in part by JST ACCEL Grant Number JPMJAC1602, Japan.

REFERENCES

1. Linas Baltrunas, Marius Kaminskas, Bernd Ludwig, Omar Moling, Francesco Ricci, Aykan Aydin, Karl-Heinz Lüke, and Roland Schwaiger. 2011. InCarMusic: Context-Aware Music Recommendations in a Car. In *Proceedings of the 12th International Conference on E-Commerce and Web Technologies*. 89–100.

2. Davide Canali, Andrea Lanzi, Davide Balzarotti, Christopher Kruegel, Mihai Christodorescu, and Engin Kirda. 2012. A quantitative study of accuracy in system call-based malware detection. In *Proceedings of the 21st International Symposium on Software Testing and Analysis*. 122–132.

3. João Paulo V. Cardoso, Luciana Fujii Pontello, Pedro H. F. Holanda, Bruno Guilherme, Olga Goussevskaia, and Ana Paula Couto da Silva. 2016. Mixtape: Direction-Based Navigation in Large Media Collections. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*. 454–460.

4. Anveshi Charuvaka and Huzefa Rangwala. 2015. 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*. 675–690.

5. Matthew L. Cooper and Jonathan Foote. 2002. Automatic Music Summarization via Similarity Analysis. In *Proceedings of the 3rd International Society for Music Information Retrieval Conference*. 81–85.

6. Koby Crammer, Alex Kulesza, and Mark Dredze. 2013. Adaptive regularization of weight vectors. *Machine Learning* 91, 2 (2013), 155–187.

7. Roger B. Dannenberg and Masataka Goto. 2008. *Music Structure Analysis from Acoustic Signals*. Springer New York, New York, NY, 305–331.

8. Andrew Demetriou, Martha Larson, and Cynthia C. S. Liem. 2016. Go with the Flow: When Listeners Use Music as Technology. In *Proceedings of the 17th International Society for Music Information Retrieval Conference*. 292–298.

9. James Fogarty, Scott E. Hudson, Christopher G. Atkeson, Daniel Avrahami, Jodi Forlizzi, Sara B. Kiesler, Johnny C. Lee, and Jie Yang. 2005. Predicting human interruptibility with sensors. *ACM Transations on Computer-Human Interaction* 12, 1 (2005), 119–146.

10. Masataka Goto, Kazuyoshi Yoshii, Hiromasa Fujihara, Matthias Mauch, and Tomoyasu Nakano. 2011. Songle: A Web Service for Active Music Listening Improved by User Contributions. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*. 311–316.

11. Masataka Goto, Kazuyoshi Yoshii, and Tomoyasu Nakano. 2015. Songle Widget: Making Animation and Physical Devices Synchronized with Music Videos on the Web. In *Proceedings of the 2015 IEEE International Symposium on Multimedia*. 85–88.

12. Susan Hallam, John Price, and Georgia Katsarou. 2002. The Effects of Background Music on Primary School Pupils' Task Performance. *Educational Studies* 28, 2 (2002), 111–122.

13. Masahiro Hamasaki and Masataka Goto. 2013. 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*. 4:1–4:10.

14. Masahiro Hamasaki, Hideaki Takeda, and Takuichi Nishimura. 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*. 165–168.

15. Rong Hwa Huanga and Yi Nuo Shih. 2011. Effects of background music on concentration of workers. *Work* 38, 4 (2011), 383–387.

16. Shobu Ikeda, Kenta Oku, and Kyoji Kawagoe. 2016. Music Playlist Recommendation Using Acoustic-Feature Transitions. In *Proceedings of the 9th International C* Conference on Computer Science & Software Engineering*. 115–118.

17. Carlos Nascimento Silla Jr. and Alex Alves Freitas. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery* 22, 1-2 (2011), 31–72.

18. Hao Liu, Jun Hu, and Matthias Rauterberg. 2010b. iHeartrate: a heart rate controlled in-flight music recommendation system. In *Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research*. 26:1–26:4.

19. Ning-Han Liu, Shu-Ju Hsieh, and Cheng-Fa Tsai. 2010a. An intelligent music playlist generator based on the time parameter with artificial neural networks. *Expert Systems with Applications* 37, 4 (2010), 2815–2825.

20. Beth Logan and Stephen M. Chu. 2000. Music summarization using key phrases. In *Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing*. 749–752.

21. Adam J. Lonsdale and Adrian C. North. 2011. Why do we listen to music? A uses and gratifications analysis. *British Journal of Psychology* 102, 1 (2011), 108–134.

22. Nuria Oliver and Fernando Flores-Mangas. 2006. MPTrain: a mobile, music and physiology-based personal trainer. In *Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services*. 21–28.

23. Nuria Oliver and Lucas Kreger-Stickles. 2006. PAPA: Physiology and Purpose-Aware Automatic Playlist Generation. In *Proceedings of the 7th International Society for Music Information Retrieval*. 250–253.

24. Elias Pampalk, Tim Pohle, and Gerhard Widmer. 2005. Dynamic Playlist Generation Based on Skipping Behavior. In *Proceedings of the 6th International Society for Music Information Retrieval*. 634–637.

25. Frances H Rauscher, Gordon L Shaw, and Catherine N Ky. 1993. Music and spatial task performance. *Nature* 365, 6447 (1993), 611–611.

26. Konrad Rieck, Philipp Trinius, Carsten Willems, and Thorsten Holz. 2011. Automatic analysis of malware behavior using machine learning. *Journal of Computer Security* 19, 4 (2011), 639–668.

27. E. Glenn Schellenberg, Takayuki Nakata, Patrick G. Hunter, and Sachiko Tamoto. 2007. Exposure to music and cognitive performance: tests of children and adults. *Psychology of Music* 35, 1 (2007), 5–19.

28. Yading Song, Simon Dixon, and Marcus Pearce. 2012. A Survey of Music Recommendation Systems and Future Perspectives. In *Proceedings of the 9th International Symposium on Computer Music Modeling and Retrieval*. 395–410.

29. Takahiro Tanaka and Kinya Fujita. 2011. Study of user interruptibility estimation based on focused application switching. In *Proceedings of the 2011 ACM Conference on Computer Supported Cooperative Work*. 721–724.

30. Yoshisuke Tateyama, Yoshio Matsumoto, and Satoshi Kagami. 2004. Concentration detection by eye movements: towards supporting a human. In *Proceedings of the 2004 IEEE International Conference on Systems, Man & Cybernetics*. 1544–1548.

31. George Tzanetakis and Perry Cook. 2000. MARSYAS: a framework for audio analysis. *Organised Sound* 4 (12 2000), 169–175. Issue 03.

32. Aäron van den Oord, Sander Dieleman, and Benjamin Schrauwen. 2013. Deep content-based music recommendation. In *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*. 2643–2651.

33. Xinxi Wang, David S. Rosenblum, and Ye Wang. 2012. Context-aware mobile music recommendation for daily activities. In *Proceedings of the 20th ACM Multimedia Conference*. 99–108.

34. Alan Wells. 1990. Popular Music: Emotional Use and Management. *The Journal of Popular Culture* 24, 1 (1990), 105–117.

35. Manuela Züger and Thomas Fritz. 2015. Interruptibility of Software Developers and its Prediction Using Psycho-Physiological Sensors. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2981–2990.