공학석사학위논문

대조학습을 통한 콘텐츠 기반 음악 추천에서의 비선호도 반영

Exploiting Negative Preference in Content-based Music Recommendation with Contrastive Learning

2023년 2월

서울대학교 융합과학기술대학원
지능정보융합학과
박 민 주

공학석사학위논문

대조학습을 통한 콘텐츠 기반 음악 추천에서의 비선호도 반영

Exploiting Negative Preference in Content-based Music Recommendation with Contrastive Learning

2023년 2월

서울대학교 융합과학기술대학원
지능정보융합학과
박 민 주

대조학습을 통한 콘텐츠 기반 음악 추천에서의 비선호도 반영

Exploiting Negative Preference in Content-based Music Recommendation with Contrastive Learning

지도교수 이 교 구

이 논문을 공학석사 학위논문으로 제출함

2023년 2월

서울대학교 융합과학기술대학원

지능정보융합학과

박 민 주

박민주의 공학석사 학위 논문을 인준함

2023년 2월

위 원 장:	이	원	종	(인)
부위원장:	이	교	구	(인)
위 원:	서	봉	원	(인)

Abstract

Advanced music recommendation systems are being introduced along with the development of machine learning. However, it is essential to design a music recommendation system that can increase user satisfaction by understanding users' music tastes, not by the complexity of models. Although several studies related to music recommendation systems exploiting negative preferences have shown performance improvements, there was a lack of explanation on how they led to better recommendations.

In this work, we analyze the role of negative preference in users' music tastes by comparing music recommendation models with contrastive learning exploiting preference (CLEP) but with three different training strategies - exploiting preferences of both positive and negative (CLEP-PN), positive only (CLEP-P), and negative only (CLEP-N). We evaluate the effectiveness of the negative preference by validating each system with a small amount of personalized data obtained via survey and further illuminate the possibility of exploiting negative preference in music recommendations. Our experimental results show that CLEP-N outperforms the other two in accuracy and false positive rate. Furthermore, the proposed training strategies produced a consistent tendency regardless of different types of front-end musical feature extractors, proving the stability of the proposed method.

주요어: content-based music recommendation, negative preference, contrastive learning

학번: 2021-24997

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Nowadays, the music industry is dominated by streaming services such as Spotify and Apple Music. As it is seen in Figure 1.1, the number of users of digital music streaming services is increasing every year and their popularization has made a lot of music accessible to people. People have the advantage of having so many options for music to listen to, but on the other hand, it also brings difficulty in deciding which to listen. Music streaming services are introducing music recommendation technology to help each consumer with it, and the recommendation technology is being closely linked to the competitiveness of the service. Personalized music recommendation technology has become an essential factor for both users and online music streaming services. The need and interest in music recommendations have increased, and many related studies have been conducted actively in recent years. Various hybrid recommendation methods have been proposed, led by the collaborative filtering methods in which recommendations are made based on the user's listening history and the content-based filtering methods which are based on the song's content. The methods also differ by the defined recommendation task depending on the specific interest of

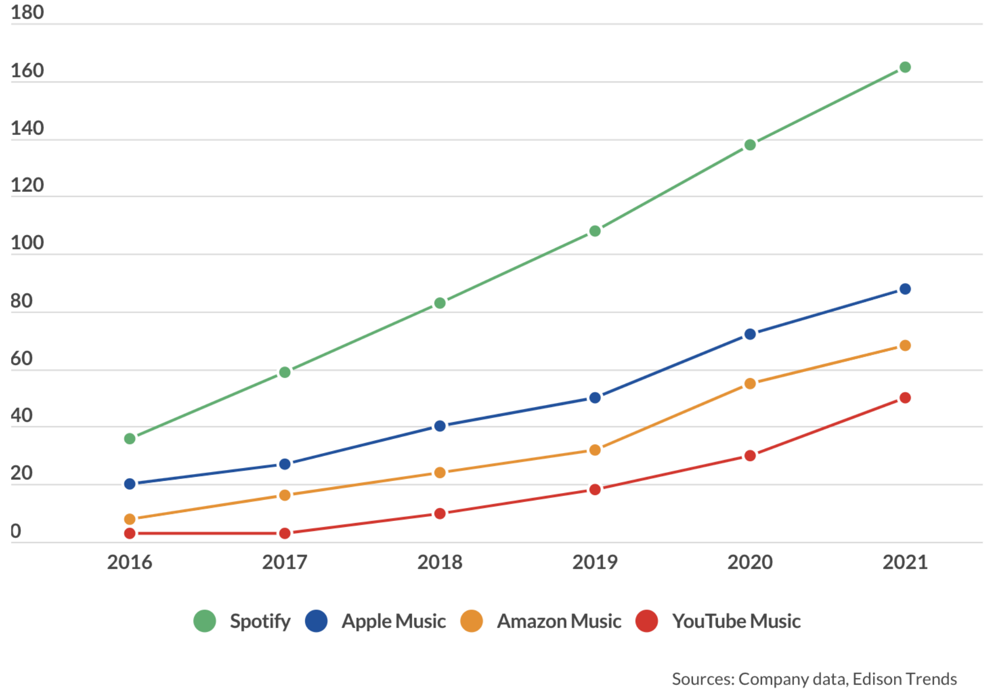


Figure 1.1: Music streaming subscribers by app 2016 to 2021 (million) [1]

recommendation. Based on the user's usage history, you can recommend songs that users will like, or recommend songs that will follow when a playlist is given.

As machine learning technology is rapid advance, music recommendation technologies applying new deep learning models are continuously being proposed. However, recent studies on music recommendation systems were mainly conducted to improve performance by adding new features or using novel machine learning techniques such as latent factor models or deep representation learning [2]. There is no doubt that improving the model's performance is essential, but fundamental analysis of why better recommendations have become possible is sometimes overlooked due to the focus on the engineering perspective. [3] pointed out that recent advances in more complex recommendation models further brought the difficulty of transparency. Recommendation is all about personalization, and its objective is to model one's music taste. It is supported by [4], indicating that the actual goal of most real-world recommendation systems is ”to influence the user to consume more items than she would have without the recommendations, not to predict the next item the user will consume.” In order to systematically explain the mechanisms of improvement in recommendation systems, one's music taste ought to be understood.

In order to explain an individual's taste in music, the following two questions must be answered - ”What kind of music do I like?” and ”What kind of music do I hate?” The naïve idea that motivated our work is, "Isn't it easier to explain the music I hate than the music I like?" Several studies have shown better performance when implementing negative feedback in recommendation systems. [5, 6] applied negative feedback in music recommendations, but they focused on proposing new architecture designs considering the negative feedback as an additional feature. Our work goes beyond simply proposing a recommendation system to which negative feedback is applied and aims to illustrate the role of negative feedback in modeling music taste. In a way, one's music taste can be seen as a set of pairs consisting of songs and corresponding preferences. Thus the song's content must be considered in order to approach the concept of music taste.

Nevertheless, there has been no attempt to explain music taste by applying positive and negative feedback to content-based music recommendation systems. In our work, we will apply negative feedback based on the content-based filtering method and explain some parts of music taste focusing on negative feedback. To prevent the ambiguity of negative feedback and naturally relate it to the concept of music taste, we will use the term ”negative preference” by borrowing the expression of [5].

1.2 Research Questions

We introduce three content-based music recommendation systems with differently conditioned contrastive learning exploiting preference (CLEP), designed based on Siamese Neural Network (SNN) [7]. The three models differ in the process of computing the final embedding vectors of the songs according to the targeted preferences - model exploiting both positive and negative preferences (CLEP-PN), model exploiting positive preference only (CLEP-P), and model exploiting negative preference only (CLEP-N). CLEP-PN embeds songs in a way that both positive and negative preferences are characterized. CLEP-P embeds songs in a way that positive preference is solely characterized, and CLEP-N embeds songs in a way that negative preference is solely characterized. Three different representations for each song will then be obtained from the frozen networks and will be used to train a simple classifier to match the preferences of each song. Afterward, the models are trained to fit a single user to fully analyze the effect of personal preferences.

We generated a user preference dataset via survey, consisting of pairs of songs and corresponding preferences for every user, obtained from twenty-four participants. The models are then trained with each dataset to predict the participant's preference for a new song. For the training, the features of each song are represented using existing works of musical feature extraction. To guarantee the stability of our work, we used three different musical feature extraction methods - Contrastive Learning of Musical Representations (CLMR) [8], Music Effects Encoder (MEE) [9], and Jukebox [10]. Our models will finally be evaluated on the test set with accuracy, precision, recall, area under the receiver operating characteristic curve (AUROC), and false positive rate. By comparing these metrics of the three models, we will be able to understand the effects of preferences, especially negative preferences, and further explain the relevant parts of music taste.

Throughout our work, we will be investigating the following research questions.

- **RQ 1.**

  What characteristics do negative preferences have in terms of explaining music taste?

  We will identify that compared to positive preference, negative preference in music taste has more distinct characteristics and that it is easier to explain music taste through negative preference.

- **RQ 2.**

  How does applying negative preference help improve music recommendations?

  We will discuss the advantages of exploiting negative preference in music recommendations through the identified roles of negative preference in music taste.

Chapter 2

Background

In this section, we provide an overview of content-based music recommendations and previous attempts to exploit negative preferences in recommendation systems. This chapter explains the theory behind our work and the field of related studies. The 2.1 briefly explains recommender systems and contrastive learning, which are essential for understanding our work. 2.2 illustrates the related preliminary studies, along with their significance and limitations.

2.1 Background Theories

2.1.1 Recommender Systems

According to the definition of [11], recommender systems are software tools and techniques providing suggestions for items to be a use to a user. The vast growth of various online services have resulted into a surge in the number of service users and the items available. Regarding the huge pool of items, finding an appropriate item might be searching a needle in a haystack. Recommender systems support the process, helping users decide which item to consume. Non-personalized recommendations can be conducted by recommending top 10 popular items, but these types of recommender systems are not mostly a big interest in recommender system research. Personalized recommender systems which aim to predict the suitable items for each user are in the main focus, and we will be referring personalized recommender systems with the term 'recommender systems' throughout our paper.

Recommender system is a very important technology that determines the competitive edge of online services, and is being actively developed through various methods. Based on the different approaches, [12] has distinguished recommender systems into six different classes - collaborative filtering, content-based, community based, demographic, knowledge-based, and hybrid recommender systems. Among these, collaborative filtering method which is commonly used in the field and content-based method which will be used as the main method in the paper will be discussed in this chapter.

- **Collaborative Filtering**

  Collaborative filtering method bases on user logs. Past histories are used to compute the user's taste, and items that are liked by other users with similar taste are recommended. Given a database of users, items, and the users' log of previously consumed items, the system can focus on either the user or the item, which is called user-based collaborative filtering and item-based collaborative filtering. User-based method predicts a user's rating for an unseen item by referring the logs of other users that have similar preferences. Meanwhile, in the same situation, item-based method refers to other items with similar ratings. Due to its strength that it can be applied regardless of the domain of the item, collaborative filtering is known as the most widely implemented technique in recommender systems.

However, there are some drawbacks of collaborative filtering. ”Cold start” problem is one of the most well known problem of collaborative filtering. Since it is being implemented based on the usage history, newly entered users and items cannot be applied. Also, ”long tail” problem, or ”popularity bias”, is another problem that can occur. Since recommendations are made according to the history of items consumed, it is more likely that popular items are often recommended. Due to this trend, items with a small history of consumption continue to be excluded from recommendation, resulting in a bias according to the item’s popularity.

- **Content-Based**

  Content-based recommendation focuses on the features of the items. By computing the similarity between items, content-based recommender systems recommend items that are similar with the previously liked items. Since the similarity is computed with the extracted set of features of items, depending on the item domain, this method may or may not be convenient. For example, content-based recommendation is generally used in text-based items, which can be represented with text-based keywords. On the other hand, these methods can be difficult to apply for domains where content is difficult to analyze or where information retrieval technology for related fields is not sufficiently advanced. Also, since the similarity of the content is not directly related to the user’s preference, there exists a semantic gap between them. Another limitation of content-based recommendation is that there is a lack of novelty due to its high reliance on the similarity of contents, continuously recommending similar items.

2.1.2 Music Recommendation System

Along with the development of the recommender system, the music recommendation system has also developed in various ways. For example, Last.fm [1] which is a famous music recommendation site recommends music that users will find interesting based on the songs they have listened to. In addition, music recommendations are also provided in music streaming services such as Spotify and Apple Music, helping people decide what to listen to. In this section, we will introduce various approaches to the music recommendation system and datasets that can be used for music recommendation research.

Approaches

The music recommendation systems follow the classification of the general recommendation system introduced above. Collaborative filtering and content-based methods are the representative recommendation methods, and they have characteristics suitable for the domain of music. Music is only consisting of audio and has shorter duration compared to other domains, and these all lead to the characteristics of the music recommendation system.

Although the operation method of collaborative filtering is independent of the characteristics of the domain, there may be differences in performance depending on the amount of data and sparsity. In the case of music, the consumption history will be significantly higher than in other domains due to its short consumption time. Hence, data handling is especially important in using collaborative filtering.

In the case of the content-based method, there is a difference in methodology depending on how the content of music is handled. With the development of music infor-

[1]http://www.last.fm

mation retrieval theory, various methods of handling music have been proposed, and studies utilizing them for music recommendation have been proposed.

Since there are both advantages and disadvantages in the methods of music recommendation, the hybrid method, which uses various methods together, is known to be most actively used in commercial music streaming services. In addition, a study on contextual music recommendation aimed at recommending music suitable for context such as mood and location is also being conducted.

Benchmark Dataset

For music recommendation research, it is important to select a dataset with various information. However, it is not easy to produce a large dataset due to user privacy and music copyright. Nevertheless, there are several published benchmark datasets, and some of them are following.

- **Million Song Dataset**
  Million Song Dataset is a cluster of dataset containing the data for one million songs and following users, provided by 'The Echo Nest'. Analyzed features such as loudness, danceability measures are included and metadata such as genre, tags are also part of the dataset. Million Song Dataset is the most widely used data in the field of music information retrieval or music recommendation research because it is consisting various information. Although the full audio file is not provided, researches on content-based music recommendation were conducted using the included features.However, there is a limit in representing the content of the song only with the provided measures, and controllability is insufficient in conducting content-based music recommendation research. A code to obtain a sample audio is provided, but it is unfeasible since the service is currently suspended.

- **The Million Playlist Dataset**

  Spotify Research, which is actively developing music recommendation technology, has also released a large dataset for music recommendation research - The Million Playlist Dataset. Spotify considers their users' music consuming behavior based on playlists, such as producing and listening to playlists, very important. The Million Playlist Dataset is also provided on a playlist basis, and includes 1,000,000 playlist titles created by Spotify users and song information contained therein. It is often used in the task of predicting songs that will follow from the playlist when the title of the playlist or several songs in the playlist are given.

2.1.3 Contrastive Learning

Contrastive learning is a machine learning technique which learns an embedding space where similar data pairs stay close together while dissimilar pairs far apart. It is mainly being applied in self-supervised learning since [13] proposed a framework for contrastive learning applying for unlabeled visual representations. Regarding identical samples with different augmentation operators positive samples, employing contrastive learning successfully helps extracting the representations of unlabeled data samples. The objective of contrastive learning is visualized in Figure 2.1.

In addition, [14] proposed SupCon, utilizing the framework of contrastive learning in supervised manner, using labeled dataset. In the existing self-supervised contrastive learning, only samples that were augmented from one sample are considered positive samples. On the other hand, SupCon considers different samples with the same label and their augmented data samples as positive samples. Accordingly, contrastive learning is being performed with additional information of data labels. The differences of the model architectures are visualized in Figure 2.2.

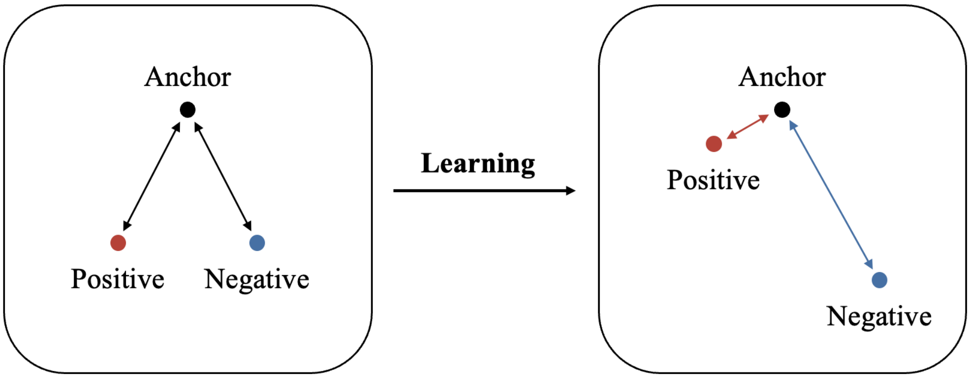
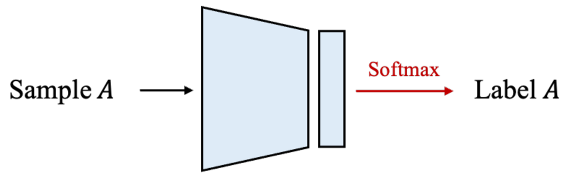


Figure 2.1: Overall scheme of contrastive learning

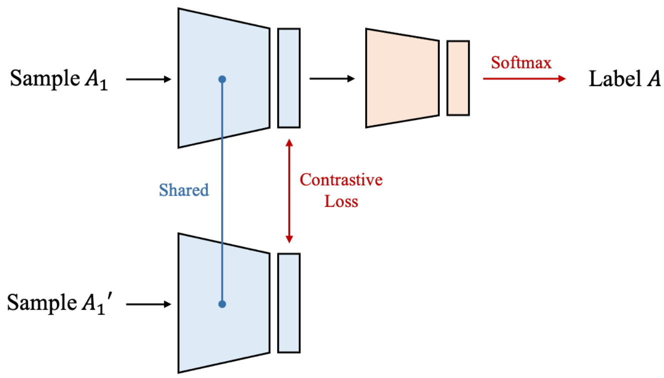
2.2 Related Works

2.2.1 Content-based Music Recommendation

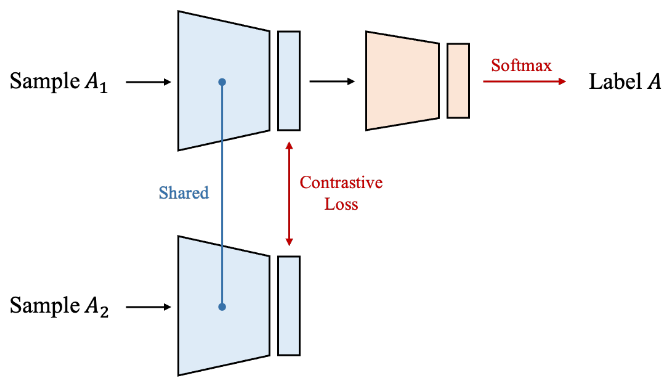
Content-based music recommendation systems have a strong advantage in that the audio content itself is utilized. Due to its reliance on the content, it compensates for the limitations of collaborative filtering methods, such as the cold-start problem, which is a problem caused by a deficiency in the information about new items or new users. Traditional content-based music recommendation systems are mainly based on meta-data such as artists, albums, or genres. However, developments in music information retrieval have facilitated the handling of music content in various ways. It became possible for both high-level audio features (e.g., melody, harmony, rhythm) and low-level audio features (e.g., Mel-Frequency Cepstral Coefficients (MFCC), mel-spectrogram) to be used for music representation. Furthermore, musical feature extractors utilizing advanced deep learning techniques have also been proposed. [8] and [9] use the idea of contrastive learning, and [10] use the idea of multi-scale VQ-VAE to extract low-level features of music.



(a) Supervised Learning



(b) Self-Supervised Contrastive Learning



(c) Supervised Contrastive Learning

Figure 2.2: Model architecture of (a) supervised learning (b) self-supervised contrastive learning (c) supervised contrastive learning

Accordingly, content-based music recommendations adopting these features are being suggested, demonstrating their practical applicability [15]. In addition to using various methods in extracting musical features, deep learning techniques based on simple front-end features are used to predict music's latent factors which can be utilized for content-based music recommendations. [16] proposed a latent factor model which maps mel-spectrogram to the item latent factor vectors obtained from the collaborative filtering method using deep convolutional neural networks.

Based on different representations, content-based music recommendation systems compute the distances between songs and recommend songs similar to the ones the user likes. The computation methods of distance also vary by model [17], but the point to note is that content-based music recommendation systems usually rely on similarity. High reliance on similarity causes the recommendations to lack novelty, and content similarity was once criticized for not being able to completely capture the preferences of a user [18]. We expect to overcome these problems by exploiting user preference data along with the contents, referring to the work of [16] which successfully bridges the semantic gap in the content-based filtering method by using the ground truth, which includes user feedback information.

2.2.2 Recommendation Systems Exploiting Negative Preference

Recommendation systems rely on user feedback, which can be divided into explicit feedback (e.g., ratings) and implicit feedback (e.g., browsing history, purchase history) according to how it is provided. Implicit feedback outnumbers explicit feedback due to its continuous update, but unfortunately, implicit feedback has a constraint that it is mainly focused on positive feedback [19, 20]. Thus modern recommendation systems are predominantly based on positive feedback, followed by the concern of its deficiencies in discriminatory power [21, 22]. In this regard, several studies are attempting to exploit negative feedback, or ”negative preference” in our term, in recommendation systems [5, 6]. [5] applied negative preference in group recommendation, showing that negative preference helped groups find consensus solutions satisfactory to all individuals. They introduced a recommendation system of avoiding the item user does not want rather than recommending the item user wants and raised the possibility of applying negative preference in recommendations.

Recommendation systems using positive and negative preferences were proposed in various domains, applying different learning models. For instance, [23, 24] exploited both positive and negative preferences by modifying graph-based recommendation systems. Proposed models have commonly shown that the user's negative preferences have increased the quality of recommendations [25,26]. For music recommendations, negative preferences are applied through skipping behaviors. [6] introduced a heuristic of automatic playlist generation by eliminating songs similar to the skipped songs. Studies covering sequential skip prediction tasks [27–29] also imply the possibilities of exploitation of negative preference in music recommendations.

Research in this field constantly mentions negative preferences, but most conclude by proposing novel architecture designs with increased performance. In contrast to these related works, we expect to focus on illuminating the specific roles of negative preferences compared to positive preferences.

Chapter 3

Methods

The main goal of our study is to understand the effects of negative preference through a comparison of recommendation models in which preferences are differently conditioned. For methodical investigation, we designed the framework to which our model will be applied consisting of three parts: feature extraction, embedding with CLEP, and preference prediction. The framework overview is visualized in Figure 3.1.

3.1 Feature Extraction

Extracting the features of songs is essential in content-based music recommendations. As described in Section 2.1, previous studies have introduced content-based music recommendation systems using various features. Considering that our work attempts to relate the contents with user feedback, we use low-level features following [16]. As representation learning has been actively studied in recent years, [8–10] proposed models trained in a self-supervised manner that produce novel music representations. There are also front-end models used in automatic music tagging [30,31], but they are models trained to classify music into a limited, discrete range of descriptions. These tags can help express the music users accept, but a much more intricate approach

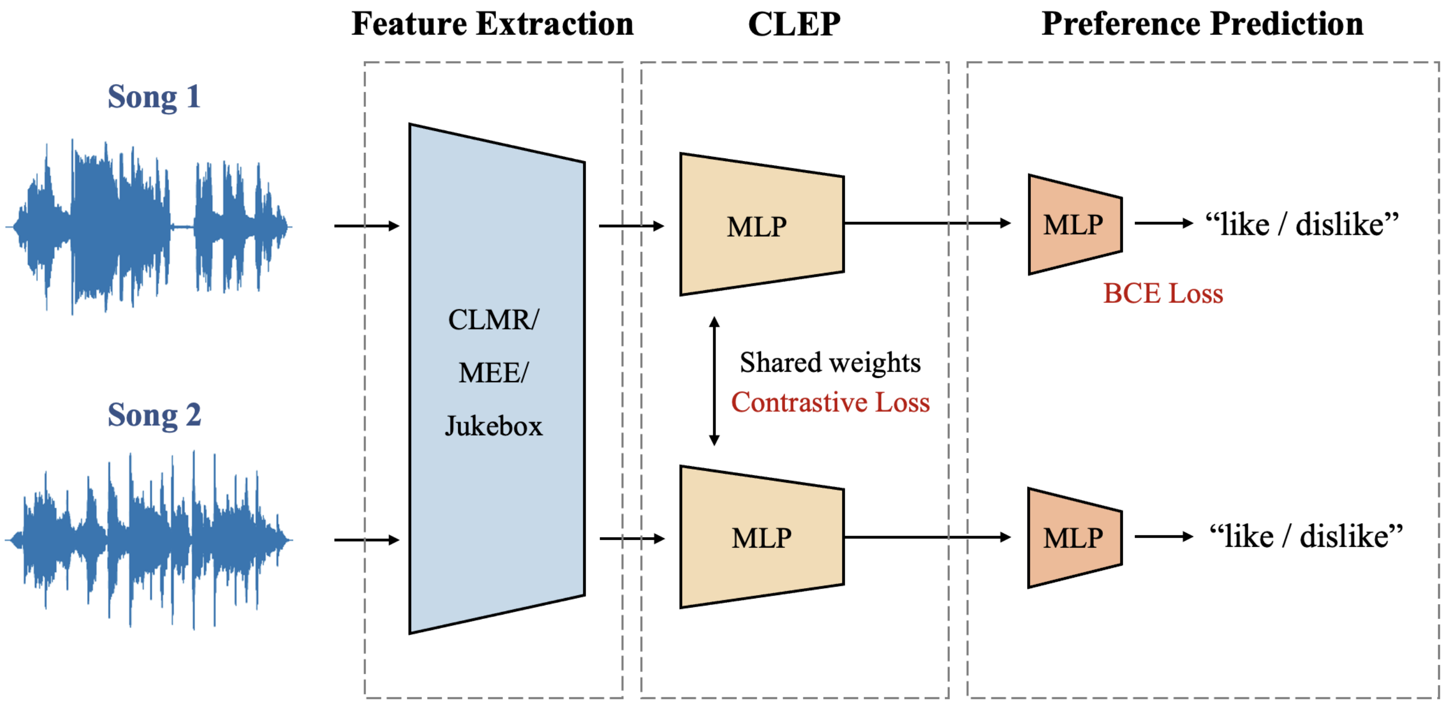


Figure 3.1: Overview of the proposed method consisting of Feature Extraction, CLEP, and Preference Prediction stage

to representation is required to relate to their music preferences. Therefore, rather than tag-based models, models based on self-supervised contrastive learning can be considered appropriate for this study since it is trained to focus on the identity of the music content itself.

By taking advantage of the pre-trained self-supervised models, the training procedure can be eased as we only need to train back-end models using a small amount of preference data of a single person. We use different feature extractors to evaluate the stability of our proposed method despite its varying performance according to each music representation. Our work uses the framework of CLMR [8], MEE [9], and Jukebox [10] as front-end musical feature extractors.

3.1.1 Contrastive Learning of Musical Representations

CLMR is a method of extracting musical features based on the idea of SimCLR [13], which performs contrastive learning by designating different sections of the same song as positive samples and sections of different songs as negative samples. In more details, to designate the positive samples, random fragment is selected from a raw audio waveform and comprehensive chain of audio augmentation is applied stochastically. Augmented samples from different raw audio waveform are identified as negative samples. Then by extracting the features using the SampleCNN architecture as the encoder, the elaborate representations are obtained through contrastive learning. To evaluate the quality of the obtained representations, music classification tasks were done with MagnaTagATune [32] and Million Song datasets [33]. Comparing with other fully supervised state-of-the-art models in tag prediction tasks, CLMR has outperformed the others.

3.1.2 Music Effects Encoder

MEE is an encoder used in a study that proposes a music remastering system. In order to capture the music's mastering style, the representation of music is extracted with a self-supervised manner similar to CLMR. MEE differs from CLMR in the architecture of the encoder and several training details. The model reproduces a mastering style similar to the target sample, indicating that the MEE successfully extracts representations that imply the characteristics of the music.

3.1.3 Jukebox

Unlike the previous two models, Jukebox is model proposed for music generation. Jukebox introduces Music VQ-VAE using the architecture of hierarchical VQ-VAE [34, 35]. Among various music generation models, Jukebox is well known for generating high-quality music with high controllability. It can be inferred that musics are represented with latent vectors that reflect their characteristics. Hence, we will use the encoder part of the Music VQ-VAE as one of our feature extractors.

Thanks to the provision of pre-trained models, we take each model to extract the features of songs for our work. Details of the musical feature extraction models are illustrated in Table 3.1. The amount of data was too small to show statistically significant results by training the front-end model from scratch, and it was shown as we expected in preliminary experiments - training a simple convolutional neural network (CNN) with mel-spectrogram input and training a network adopting the idea of CLMR. Therefore, we will be focusing on the pre-trained musical feature extractors for the rest of our work.

Front-end Models	Dataset (# Tracks)	Sampling Rate, Channel	Dimension
CLMR	MagnaTagATune [32] (187k)	16 kHz, mono	512
MEE	MTG-Jamendo [36] (55k)	44.1kHz, stereo	2048
Jukebox	web crawled (1.2m)	44.1kHz, mono	4800

Table 3.1: Details of the front-end musical feature extraction models

3.2 Contrastive Learning Exploiting Preference (CLEP)

We devise three different content-based music recommendation models as follows:

- **CLEP-PN**
  Model with contrastive learning exploiting both positive and negative preferences
- **CLEP-P**
  Model with contrastive learning exploiting positive preference only
- **CLEP-N**
  Model with contrastive learning exploiting negative preference only

The three models are differentiated in the embedding part. The representations obtained in the previous part are embedded considering the preferences using the architecture of SNN. SNN learns representations by adjusting the distance between the embeddings according to the labels of item pairs. In more detail, SNN is trained with contrastive loss as follows:

$$L^{Contrastive} = yD^2 + (1-y)max(margin - D, 0)^2 \tag{3.1}$$

where y is the label of an item pair and D is the distance between the items. When

Data Pair	Pos - Pos	Neg - Neg	Pos-Neg
CLEP-PN	y=1	y=1	y=0
CLEP-P	y=1	y=0	y=0
CLEP-N	y=0	y=1	y=0

Table 3.2: Designated labels of data pairs in the contrastive learning phase according to the proposed models

a pair of items labeled as $y = 1$ is given, it leads to $L = D^2$, reducing the distance as training. That is, a pair of items that is labeled as $y = 1$ will be embedded close together in the embedding space. On the other hand, when a pair of items labeled as $y = 0$ is given, it leads to $L = max(margin - D, 0)^2$. So as training continues, the distance gets close to the margin value. The margin value was set as $margin = 7$ through empirical observations so that the embeddings from both classes were well separated.

As it can be seen from the loss function, the embedding varies depending on how the item pairs are labeled. In general classification tasks, items that belong to the same class are embedded closer and those belonging to different classes are embedded farther. We changed the way of labeling according to the purpose of each model like the following. The way it is labeled is shown in Table 3.2 and visualizations of the embedding strategies of each model are depicted in Figure 3.2.

- **CLEP-PN**

  The label is set so that the songs with the same preference are embedded close to each other, and the songs with different preferences are embedded far apart. In other words, we set $y = 1$ for 'like-like' and 'dislike-dislike' pairs, and $y = 0$ for 'like-dislike' pairs.

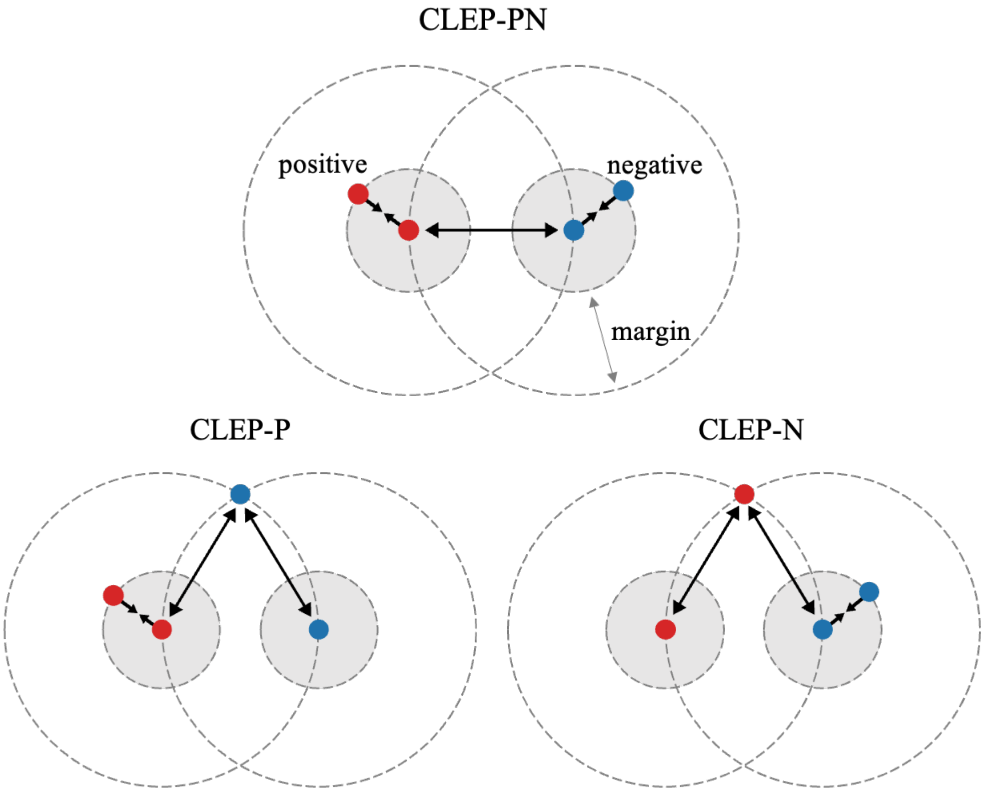


Figure 3.2: Demonstration of each embedding space of CLEP-PN, CLEP-P, and CLEP-N

- **CLEP-P**

  The label is set so that the songs with positive preferences are embedded close together, and other kinds of pairs are embedded far apart. We set $y = 1$ for 'like-like' pairs, and $y = 0$ for 'like-dislike' and 'dislike-dislike' pairs.

- **CLEP-N**

  The label is set so that the songs with negative preferences are embedded close together, and other kinds of pairs are embedded far apart. We set $y = 1$ for 'dislike-dislike' pairs, and $y = 0$ for 'like-dislike' and 'like-like' pairs.

3.3 Preference Prediction

Pre-trained musical feature extractors are often evaluated in classification tasks by appending a simple model of Multi-Layer Perceptron (MLP) [8]. We apply the same technique to predict the user's preference for each song. MLP layers are added and trained to match the ground truth of whether the user likes or dislikes the song, with Binary Cross Entropy loss (BCE loss). Then the sigmoid function eventually computes the probability of preference.

Chapter 4

Experiments

4.1 Experimental Setups

For musical feature extractors, we used the public-available pre-trained models of CLMR [1], MEE [2], and Jukebox [3]. Feature vectors of each song were extracted with the dimension denoted in Table 3.1. Sixteen songs per batch were trained with CLEP, which has a network architecture of MLP with 4, 5, and 5 layers for CLMR, MEE, and Jukebox, respectively. The preference prediction stage has a network architecture of 3-layer MLP. Both CLEP and preference prediction stage were trained using Adam optimizer with learning rate scheduled so that it is reduced when validation loss is not decreasing until two epochs. CLEP was trained for 20 epochs, and the learning rate was scheduled starting from 0.01. The preference prediction stage was trained for 30 epochs with the learning rate starting from 0.001.

[1] https://github.com/Spijkervet/CLMR

[2] https://github.com/jhtonyKoo/e2e_music_remastering_system

[3] https://github.com/openai/jukebox

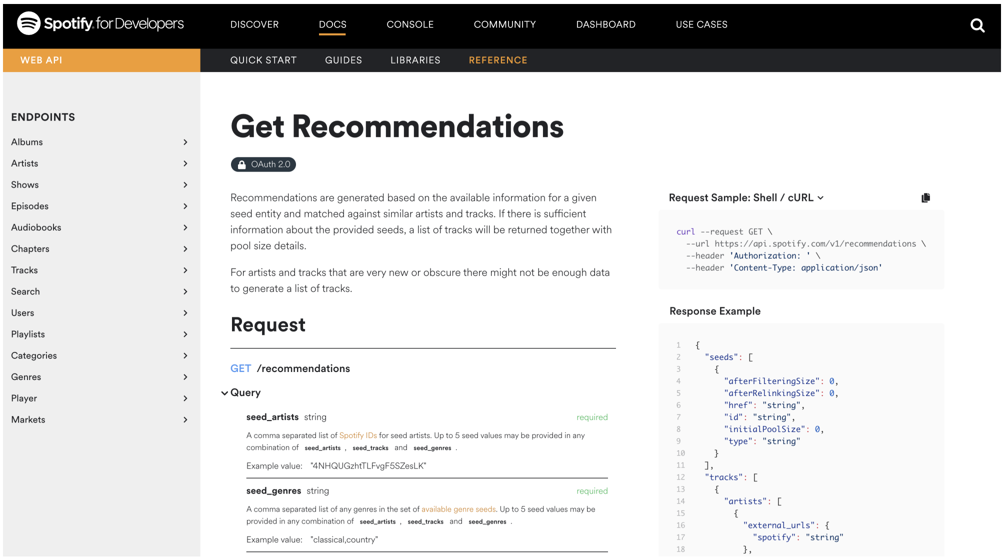
4.2 User Preference Dataset

We conducted a web-based survey asking participants about their music preferences to train and evaluate our models. It is difficult to define music preferences elaborately, but as many online music recommendation services do, user feedback can be elicited to assume their preferences [37]. The survey asked for the likes and dislikes of certain songs, and the collected data were used to represent each participant's music preference.

Twenty-four volunteers with no hearing problems were recruited from online student communities. They were all Koreans, and their ages ranged from 24 to 37, with an average of 27. After briefly introducing the survey process, we obtained consent for their participation. They were asked to listen to 200 music clips and answer whether they liked or disliked each song. Since users' familiarity with songs does affect their preference [38], 40 songs were randomly selected from different genres to reduce genre bias and effects on the popularity of the songs. They consisted of the five most popular genres nowadays - rock, EDM, hip-hop, pop, and R&B. We used the 'Get Recommendations' function provided in Spotify API [4], which can return a list of tracks when given a particular genre as shown in Figure 4.1. Music excerpts of 10 seconds were randomly selected from each track and given in random order to each participant. By referring to [38], which studied music preference and recognition, it was considered that 10 seconds were enough for the participants to identify the melodies and decide their preferences on each song. The music clips were given stereo-channeled with a sampling rate of 44.1kHz in the survey but were manipulated in the feature extraction stage to fit each feature extraction model. A part of the questionnaire can be checked in Figure 4.2.

[4] https://developer.spotify.com/documentation/web-api/

Figure 4.1: Website of "Spotify for Developers" showing the description of using Spotify API for "Get Recommendations"



Through the survey, we obtained each participant's preferences for 200 songs. Each participant had a different ratio of their liked and disliked songs - some had much more liked songs while some had much more disliked songs. Table 4.1 shows the specific number of responses according to the participants' age and preference.

The average ratio of the number of liked songs to the number of disliked songs was 0.96:1 on average, saying the preferences of the entire participants were not biased. Within the 200 individual data, we divided them into a training set and a test set at a ratio of 3:1. We then trained the models with the training set and assessed their performances on the test set.

Figure 4.2: Part of the questionnaire used for data collection

실험 안내

본 설문은 '비선호도를 반영한 콘텐츠 기반 음악 추천'에 대한 연구를 수행하기 위한 설문입니다.
조용한 환경에서 청취하는 것을 권장드립니다.
설문은 총 200개 문항으로 이루어져 있으며, 예상 소요시간은 35분입니다.
각 문항 당 10초 분량의 음악을 듣게 되실 것입니다.
음악을 들으시고 본인 취향에 맞으시다면 '선호'를, 맞지 않으시다면 '비선호'를 선택해주세요.
선호 및 비선호의 기준이 애매하시다면, '랜덤재생 시 이 음악이 나왔을 경우 계속 들어볼 것인가?'라는 질문에 대입하여 응답하셔도 좋습니다.

1/200

1. 위 곡에 대한 선호도를 선택해주세요.

- 선호
- 비선호

2/200

2. 위 곡에 대한 선호도를 선택해주세요.

- 선호
- 비선호

Participant	**Age**	**Positive**	**Negative**
1	24	101	99
2	27	62	138
3	27	47	153
4	24	49	151
5	27	87	113
6	25	54	146
7	26	65	135
8	27	97	103
9	27	78	122
10	24	29	171
11	31	70	130
12	25	75	125
13	37	175	25
14	28	64	136
15	26	45	155
16	28	78	122
17	24	141	59
18	31	66	134
19	25	24	176
20	24	63	137
21	29	35	165
22	25	125	75
23	27	129	71
24	24	88	112
Mean	**26.75**	**76.96**	**123.04**

Table 4.1: Demographics from people who participated in the data collection and the number of responses for preferences and negative preferences

4.3 Evaluation

4.3.1 Evaluation Metric

To compare the performance of each model, we used the following five metrics - accuracy, precision, recall, area under the receiver operating characteristic curve (AUROC), and false positive rate.

Accuracy

Since our evaluation task is a binary classification task, accuracy, which is the ratio of the number of correct answers to the total prediction, can be measured. If there is a model with a high accuracy, it can be interpreted that the model closely builds the embedding space containing the user's preference through contrastive learning.

Precision

Precision is the ratio of correct answers among the predicted positive samples. It is obtained by dividing True Positive by the sum of True Positive and False Positive.

Recall

Recall is the ratio of correct answers among the samples which their labeled positive. In other words, recall is the value of True Positive divided by the sum of True Positive and False Negative. Precision and Recall have a trade-off relationship that cannot be increased together, so F1-score, which is a harmonic average thereof, is also used as an evaluation index. However, in our work, the measurement of F1 score is omitted to focus on the tendency of each metric.

Area Under the Receiver Operating Characteristic Curve

The Area Under the Receiver Operating Characteristic Curve (AUROC) refers to the area below the ROC curve where the values of false positive rate and true positive rate according to various thresholds are shown. The existence of a threshold that derives a low false positive rate and a high true positive rate means that the classification performance of the model is excellent, so it can be interpreted that the larger the area under the ROC curve, the higher the performance.

False Positive Rate

False Positive Rate is the value of False Positive divided by the sum of False Positive and True Negative. The recommendation system field has plentiful evaluation methodologies [39–41], but most of them focus on true positives as the evaluation objective. However, [42] points out that false positives are a clear concern in music recommendations. From the user experience perspective, users are not aware of not being recommended a song they like. Instead, it is more disappointing to be recommended a song they dislike. Since false positives negatively affect user experience compared to false negatives, measuring false positive metrics will help analyze the practical utility of a music recommendation system. From this point of view, it is crucial to look into precision and false positive rate, which are the metrics relevant to false positives. Precision in recommendation refers to the ratio of liked songs over recommended ones. Meanwhile, the false positive rate is the ratio of recommended songs over the songs that which user truly dislikes. High precision and low false positive rate imply that the recommendation system is worthwhile in terms of user experience.

4.3.2 Experimental Results

The experimental results were analyzed through the Friedman test, and the overall test results are illustrated in Table 4.2. The following χ^2 and p-value in the table demonstrate the statistical significance that the results differ by model. The results of different musical feature extractions are also displayed in the table, showing a consistent tendency to some degree regardless of feature extractors. Accuracy, recall, and false positive rate showed statistically significant differences in all three cases, while precision and AUROC showed differences only in models using Jukebox for its feature extractor. In order to verify specified relationships between the models, Wilcoxon signed-rank tests were performed as a post-hoc analysis for accuracy, recall, and false positive rate. Table 4.3 shows multiple testing results between the models, and the relationships in which model showed the best result are verifiable. The full results are shown in Table 4.4.

Front-end Models	CLEP	Accuracy (↑)	Precision (↑)	Recall (↑)	AUROC (↑)	FPR (↓)
CLMR	CLEP-PN	0.62	0.37	0.367	0.508	0.329
	CLEP-P	0.56	0.424	**0.722**	**0.588**	0.547
	CLEP-N	**0.66**	**0.5**	0.16	0.514	**0.097**
	χ^2 (df=2)	9.621 (p=0.008**)	1.595 (p=0.451)	25.613 (p=2.74e-06***)	2.083 (p=0.353)	26.547 (p=1.72e-06***)
MEE	CLEP-PN	0.59	0.334	0.453	0.502	0.352
	CLEP-P	0.55	0.375	**0.481**	0.519	0.439
	CLEP-N	**0.61**	**0.404**	0.367	**0.538**	**0.286**
	χ^2 (df=2)	7.101 (p=0.029*)	1.916 (p=0.384)	15.475 (p=0.0004***)	1 (p=0.607)	20.609 (p=3.35e-05***)
Jukebox	CLEP-PN	0.59	0.421	0.547	0.5	0.423
	CLEP-P	0.64	0.457	**0.747**	**0.653**	0.431
	CLEP-N	**0.7**	**0.519**	0.338	0.555	**0.15**
	χ^2 (df=2)	11.5 (p=0.003**)	18.583 (p=9.22e-05***)	16.28 (p=0.0003***)	21.894 (p=1.76e-05***)	25.872 (p=2.41e-06***)

Table 4.2: Median values of accuracy, precision, recall, AUROC, and false positive rate (FPR) according to the musical feature extraction models and our models. The reported χ^2 values and their p-values are obtained with Friedman test (Statistical significance : *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$).

		CLEP-PN vs P	CLEP-P vs N	CLEP-N vs PN	Results
Accuracy (↑)	CLMR	0.065	0.002**	0.028*	CLEP-N > PN, P
	MEE	0.648	0.004**	0.016*	CLEP-N > PN, P
	Jukebox	0.016*	0.038*	0.008**	CLEP-N > P > PN
Recall (↑)	CLMR	0.031*	9.6e-05***	0.009**	CLEP-P > PN > N
	MEE	0.298	0.0003***	0.029*	CLEP-PN, P > N
	Jukebox	0.026*	3.6e-05***	0.042*	CLEP-P > PN > N
FPR (↓)	CLMR	0.066	7.6e-05***	0.001**	CLEP-N < PN, P
	MEE	0.173	0.0003***	0.03*	CLEP-N < PN, P
	Jukebox	0.82	1.9e-05***	0.001**	CLEP-N < PN, P

Table 4.3: P-values of Wilcoxon signed-rank test as a post-hoc analysis of the Friedman test above (Statistical significance : *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$). Significant order relations between the models are noted on the right side.

Table 4.4: Full results showing the number of True Positive(TP), True Negative(TN), False Positive(FP), False Negative(FN) for each participant and each model

Participant	Model	TP	TN	FP	FN
1	CLEP-PN	15	27	5	3
	CLEP-P	18	16	16	0
	CLEP-N	10	32	0	8
2	CLEP-PN	0	35	0	15
	CLEP-P	10	26	9	5
	CLEP-N	1	33	2	14
3	CLEP-PN	4	24	18	4
	CLEP-P	4	29	13	4
	CLEP-N	0	37	5	8
4	CLEP-PN	9	26	9	6
	CLEP-P	9	17	18	6
	CLEP-N	4	33	2	11
5	CLEP-PN	8	22	7	13
	CLEP-P	15	19	10	6
	CLEP-N	6	23	6	15
6	CLEP-PN	4	28	8	10
	CLEP-P	9	19	17	5
	CLEP-N	1	30	6	13
7	CLEP-PN	10	26	11	3
	CLEP-P	11	18	19	2
	CLEP-N	1	32	5	12
8	CLEP-PN	18	11	10	11
	CLEP-P	26	7	14	3
	CLEP-N	7	18	3	22

Participant	Model	TP	TN	FP	FN
9	CLEP-PN	16	15	15	4
	CLEP-P	15	15	15	5
	CLEP-N	7	23	7	13
10	CLEP-PN	4	24	19	3
	CLEP-P	4	25	18	3
	CLEP-N	0	40	3	7
11	CLEP-PN	0	32	0	18
	CLEP-P	14	20	12	4
	CLEP-N	3	29	3	15
12	CLEP-PN	11	16	16	7
	CLEP-P	16	16	16	2
	CLEP-N	5	30	2	13
13	CLEP-PN	42	1	4	3
	CLEP-P	42	1	4	3
	CLEP-N	28	2	3	17
14	CLEP-PN	5	25	13	7
	CLEP-P	10	16	22	2
	CLEP-N	3	31	7	9
15	CLEP-PN	8	23	14	5
	CLEP-P	9	18	19	4
	CLEP-N	1	34	3	12
16	CLEP-PN	0	30	2	18
	CLEP-P	16	16	16	2
	CLEP-N	6	27	5	12

Participant	Model	TP	TN	FP	FN
17	CLEP-PN	30	5	8	7
	CLEP-P	34	4	9	3
	CLEP-N	28	10	3	9
18	CLEP-PN	0	30	0	20
	CLEP-P	16	16	14	4
	CLEP-N	8	22	8	12
19	CLEP-PN	2	35	11	2
	CLEP-P	2	33	13	2
	CLEP-N	0	43	3	4
20	CLEP-PN	15	12	20	3
	CLEP-P	13	19	13	5
	CLEP-N	5	31	1	13
21	CLEP-PN	5	22	15	8
	CLEP-P	6	20	17	7
	CLEP-N	2	34	3	11
22	CLEP-PN	32	0	18	0
	CLEP-P	26	8	10	6
	CLEP-N	14	13	5	18
23	CLEP-PN	24	18	2	6
	CLEP-P	24	14	6	6
	CLEP-N	13	19	1	17
24	CLEP-PN	12	16	12	10
	CLEP-P	19	10	18	3
	CLEP-N	5	22	6	17

Chapter 5

Results and Discussion

We have trained our three models - CLEP-PN, CLEP-P, and CLEP-N - to embed the contents of songs exploiting preferences and predict the preference of unknown songs. In the training phase, each data was embedded depending on its feature and preference. We observed that the songs were embedded as expected when visualized in two dimensions using t-SNE as is seen in Figure 5.1 for instance. Songs with positive and negative preferences were clustered each in the embedding space of CLEP-PN. Furthermore, songs with positive preferences were clustered while songs with negative preferences were spread out in the embedding space of CLEP-P, and vice versa in the case of CLEP-N.

As we showed, there were statistically significant differences between the models in terms of accuracy, recall, and false positive rate. First, CLEP-N showed the highest accuracy among the three models. Although the statistical significance for the difference between CLEP-PN and CLEP-P was slightly different depending on the musical feature extractors, the accuracy of CLEP-N consistently exceeded the accuracy of the other two models. In the case of precision, models which used Jukebox as its musical feature extractor only showed a significant difference ($\chi^2(2) = 18.583, p < 0.001$),

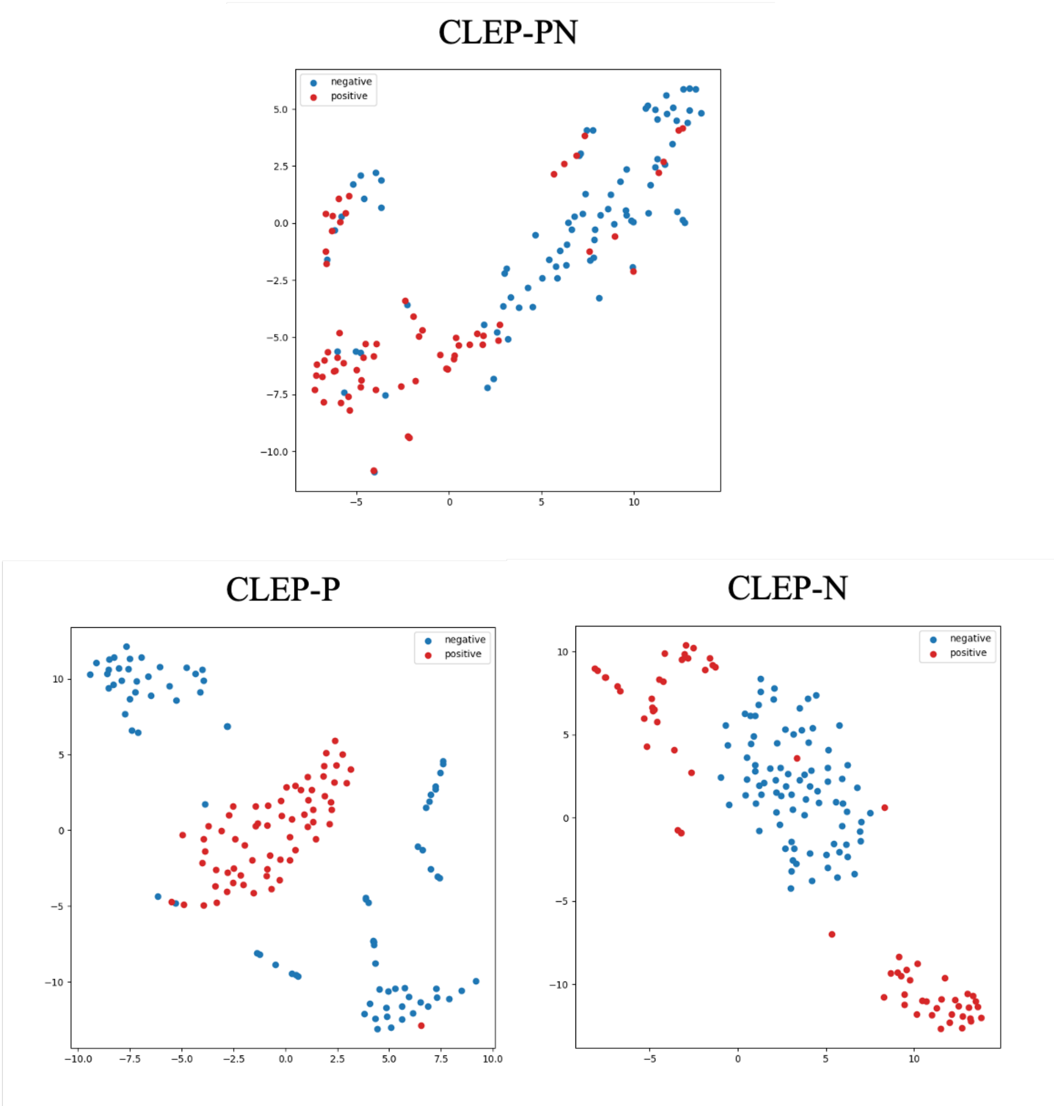


Figure 5.1: Example of t-SNE visualization of embedding spaces trained with data obtained from a single participant, with MEE as musical feature extractor. Red points represent the songs with positive preference, and blue points represent the songs with negative preference.

presenting the highest value in CLEP-N. The precision of models with other musical feature extractors showed a lack of significance, but the median values were consistently the highest in CLEP-N.

Meanwhile, the results showed that CLEP-N performed the lowest recall. In the case of false positive rate, CLEP-N outperformed the other two regardless of musical feature extractors, showing the lowest value. A recommendation system's low false positive rate implies that the model barely recommends music the user dislikes. The results of CLEP-N showing high false positive rate and low recall indicate that it is better at predicting songs the user dislikes than predicting songs the user likes. A simple approach can allow a rough guess of thinking that CLEP-N is too pessimistic, predicting that the user dislikes every song. However, considering that the survey data was balanced in terms of preferences and CLEP-N showed the highest accuracy, it is convincing enough to claim its strength. The false positive rate and recall both have actual preferences as the denominator, but the false positive rate focuses on the negatives while recall focuses on the positives. False positive rate is an anti-metric for recall, which is a metric aware of the irrelevant items returned by the recommendation systems. Considering that anti-metrics are more valuable than classical metrics when distinguishing recommendation systems with similar relevance [43], the fact that CLEP-N is showing a high false positive rate is strong evidence of its potential to be utilized in recommendation systems.

All three models showed no particular tendency in terms of AUROC, and the values were insufficient to state the stable performance of each model. As seen from the low AUROC, our models, including CLEP-N, have limitations in their immediate application as a recommendation system. It is due to the shortage of data in quantity and the simple implementation aimed at identifying the differences, and adjusting CLEP-N for real application will be left as our future work.

Based on the results, the research questions of our work as mentioned above can be discussed like the following:

- **RQ 1.**

  What characteristics do negative preferences have in terms of explaining music taste?

  If we think of a user's music taste as a complex distribution of songs the user likes and dislikes, we were interested in which of these three models most similarly simulates the distribution. If the contents of songs that the user feels positive or negative have a certain tendency, the features of the songs with the same preference will be embedded close to each other. Thus we can regard the embedding spaces of our models as the distribution of users' music tastes according to their positive and negative preferences. Based on the result that CLEP-N showed the highest accuracy, we provide evidence that songs with negative preference have more distinct characteristics than songs with positive preference. It is also supported by the concept of serendipity [44], which is a measure indicating the unexpectedness of a recommendation. The fact that users react to unexpectedly good things points out that there is a chance of finding songs the user may like in an unpredictable area of the user's music taste, and the findings of our work explain it.

- **RQ 2.**

  How does applying negative preference help improve music recommendations?

  Although our experimental settings had a gap from the real-world situation, we verified the model's potential to exploit negative preference in content-based music recommendations by conditioning the preferences in the models. From the perspective of user experience, it is shown that the model with a low false positive rate and high precision can lead the users to a pleasant experience of consuming music. Through our work, we verified that CLEP-N showed a distinctly low false positive rate and, in some cases, high precision. Therefore, we can conclude that exploiting negative preference contributes to improvement in false positive metrics, and this consideration in music recommendations will be expected to make significant progress.

Chapter 6

Conclusion

6.1 Contribution

In this work, we analyzed the role of negative preferences in users' music tastes by comparing three models with differently conditioned contrastive learning exploiting preference (CLEP) - models exploiting both positive and negative preferences (CLEP-PN), positive preference only (CLEP-P), and negative preference only (CLEP-N). We found that CLEP-N, which assumes that negative preference is more characterized, showed the highest accuracy among the three proposed models. It leads to a conclusion that negative preference has the potential to have more explainable characteristics in users' music taste compared to positive preference. Furthermore, CLEP-N outperformed the other two models in terms of false positive metrics. As false positive metrics are told as highly relevant in recommendation literature, CLEP-N also illuminates the capacity of improving music recommendations by utilizing negative preferences. Based on these results, the significance of this study will be explained in this section.

6.1.1 Novel Approach on Content-Based Music Recommendation

The previously known content-based music recommendation has chosen a method of understanding music content using audio features or tags and recommending similar songs based on them. However, as mentioned in 2.1.1, there exists a semantic gap between the similarity of the musics and user preferences. To compensate for this, it is used as a hybrid method along with collaborative filtering, which can represent the user's preference as a latent vector. Nonetheless, the two methods work as a simple ensemble, and the content of the music is not directly related to the user's preference. Since understanding the user's music preference is essential for personalized recommendation, the understanding of the user's preferred music must be supported. Our work is meaningful in suggesting a point where users' preferences and music content can be connected.

Individual music preferences are difficult to explain with simple tags. Even two different musics with the same tag or similar audio features may be liked or disliked by users due to minor differences. As can be seen in Figure 6.1, preferences and genre of music tended to be quite irrelevant within the embedded space of the learned models. In other words, it can be interpreted that the new approach proposed in our work reflects the semantic part of music preference, which is difficult to be classified as tags.

It is hoped that this will be further expanded to other domains as well as music, and actively utilize the method that can connect content and user preferences in general recommendation systems.

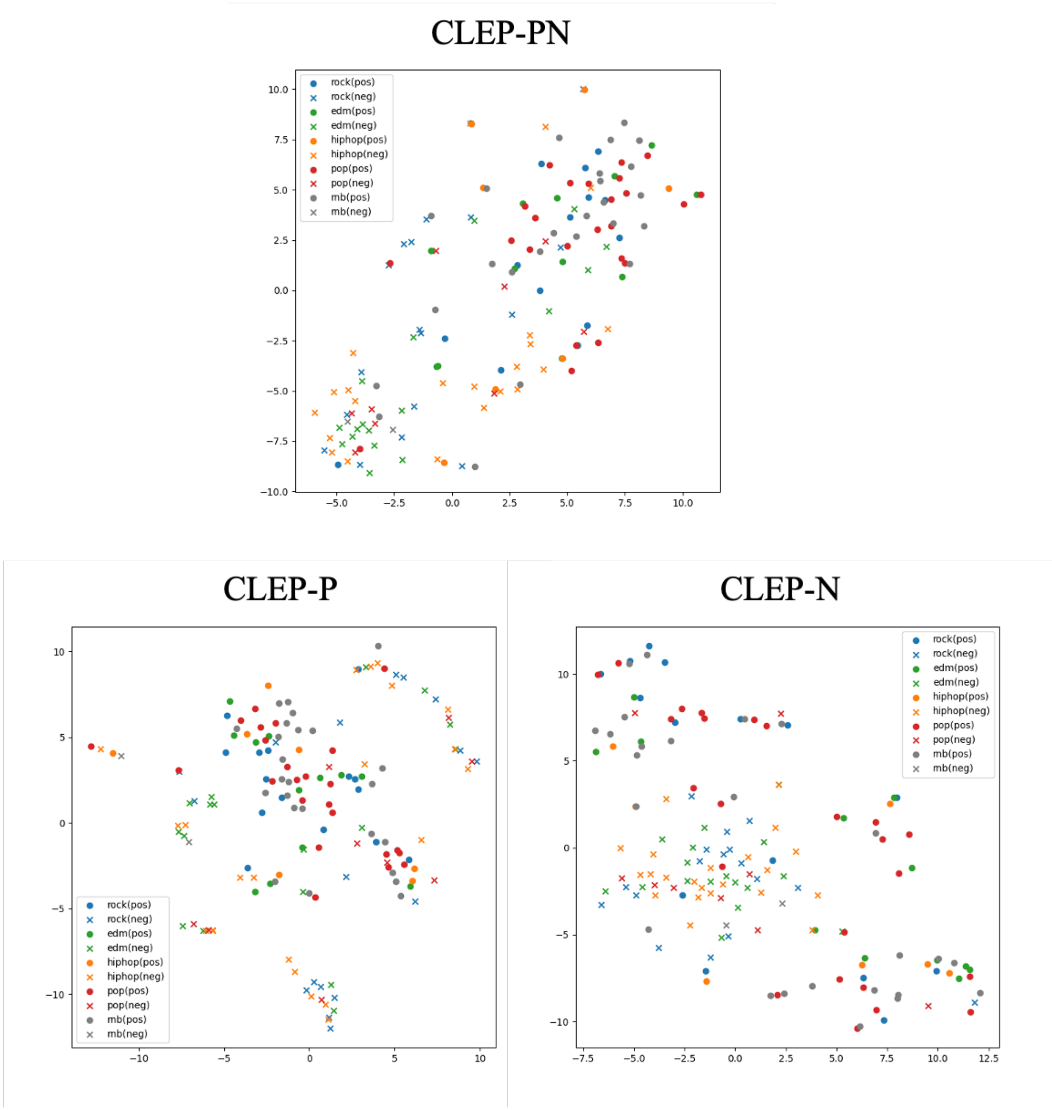


Figure 6.1: Example of t-SNE visualization of embedding spaces trained with data obtained from a single participant, with MEE as musical feature extractor. Colors are representing different genres, while shapes of the marker represent the preference. Details are notated in the legend.

6.1.2 Comprehension of Music Preference

Our work is meaningful in that it not only proposed a music recommendation method but also introduced an understanding of music preference from the results. There have been several previous studies trying to understand music preference [45, 46]. However, most of the approaches are to explain preferred music, or to analyze the factors that determine music preference rather than understanding the music preference itself [47].

Our work attempted to understand the characteristics of people's overall music preference, and presented evidence for the phenomenon that music that is not preferred over preferred music is more characteristic through experiments. It contributes to understand an individual's music preference, and more insights on music preference are expected to arise from the findings of our work.

6.2 Limitation and Future Works

We have intensified our work to enlighten the effects of negative preference through comparative analysis. In other words, our work is focused on synthesizing our novel findings for negative preferences but not on directly applicable model proposals. Therefore, there is some limitations in directly using the methodology proposed in our work for real-world applications.

Our work proposes to use the user's negative preference in music recommendation. However, it difficult to explicitly obtain the user's negative preference in most music streaming services. Although functions such as "Like" and "Hide song" exist, most people do not provide direct feedback on each music played. While people can implicitly infer negative feedback with skipping behavior, it is not accurate because people may skip musics depending on their situation or mood, regardless of negative preferences. The methodology proposed in our study has major limitation in this regard because it is based on the assumption that users have data on their preference and negative preference for music. Therefore, in order to apply the method to the actual service, methods to supplement this point must be presented.

Also, since the proposed method is learned with individual preference data, the model is trained individually. However, considering the real-world application, training the model individually for each user will require a lot of computation, resulting in inefficient scalability. Given the fact that there are numerous users and numerous songs within the music streaming service, not only providing accurate recommendations, but also operating quickly and efficiently by utilizing accessible information properly are extremely important. Therefore, based on the late vector in collaborative filtering, measures such as helping users with similar distributions of tastes to calculate will be discussed in future studies. Therefore, in order to supplement this, modifying the proposed method in utilizing the information of other users should be further studied. For instance, there may be a way to ease the training procedure among users with similar distribution of preferences, based on the latent vector from collaborative filtering. In our future work, we will consider a more generalized model training method that can cope with a vast amount of data.

There are also several limitations regarding the dataset for model training. First of all, in order to train the proposed model, music data showing individual preferences are required. However, since there is no public dataset to fit these needs, it was inevitable to produce the dataset for progressing our work. Although there were statistically significant results even if it was collected from few participants, it is still insufficient to generalize our findings. Furthermore, the fact that the preference for music was inquired in binary manner is also one of the limitations. To determine the preference of the music, different standards will be applied by person. Some people may be generous to saying that they like the song, but some may say they don't like it for a very small reason. In our work, since binary choices of 'like' and 'dislike' were demanded in the data collection process, the opinions at the boundary are not carefully reflected. In our future work, we will inquire the music preferences in more detail - in Likert scale, for example - to find out the characteristics according to the degree of preference.

Bibliography

[1] "Music streaming app revenue and usage statistics (2022)." https://www.businessofapps.com/data/music-streaming-market/. Accessed: 2022-12-05.

[2] M. Schedl, "Deep learning in music recommendation systems," *Frontiers in Applied Mathematics and Statistics*, p. 44, 2019.

[3] Y. Zhang, X. Chen, *et al.*, "Explainable recommendation: A survey and new perspectives," *Foundations and Trends® in Information Retrieval*, vol. 14, no. 1, pp. 1–101, 2020.

[4] N. Koenigstein, "Rethinking collaborative filtering: A practical perspective on state-of-the-art research based on real world insights," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pp. 336–337, 2017.

[5] D. L. Chao, J. Balthrop, and S. Forrest, "Adaptive radio: achieving consensus using negative preferences," in *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, pp. 120–123, 2005.

[6] E. Pampalk, T. Pohle, and G. Widmer, "Dynamic playlist generation based on skipping behavior.," in *ISMIR*, vol. 5, pp. 634–637, 2005.

[7] G. Koch, R. Zemel, R. Salakhutdinov, *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, p. 0, Lille, 2015.

[8] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," *arXiv preprint arXiv:2103.09410*, 2021.

[9] J. Koo, S. Paik, and K. Lee, "End-to-end music remastering system using self-supervised and adversarial training," *arXiv preprint arXiv:2202.08520*, 2022.

[10] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[11] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems: Techniques, applications, and challenges," *Recommender Systems Handbook*, pp. 1–35, 2022.

[12] R. Burke, "Hybrid web recommender systems," *The adaptive web*, pp. 377–408, 2007.

[13] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020.

[14] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18661–18673, 2020.

[15] Y. Deldjoo, M. Schedl, and P. Knees, "Content-driven music recommendation: Evolution, state of the art, and challenges," *arXiv preprint arXiv:2107.11803*, 2021.

[16] A. Van den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," *Advances in neural information processing systems*, vol. 26, 2013.

[17] N.-H. Liu, "Comparison of content-based music recommendation using different distance estimation methods," *Applied intelligence*, vol. 38, no. 2, pp. 160–174, 2013.

[18] M. Kaminskas and F. Ricci, "Contextual music information retrieval and recommendation: State of the art and challenges," *Computer Science Review*, vol. 6, no. 2-3, pp. 89–119, 2012.

[19] D. Kelly and J. Teevan, "Implicit feedback for inferring user preference: a bibliography," in *Acm Sigir Forum*, vol. 37, pp. 18–28, ACM New York, NY, USA, 2003.

[20] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli, "User profiles for personalized information access," *The adaptive web*, pp. 54–89, 2007.

[21] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *2008 Eighth IEEE international conference on data mining*, pp. 263–272, Ieee, 2008.

[22] H. Lu, M. Zhang, and S. Ma, "Between clicks and satisfaction: Study on multi-phase user preferences and satisfaction for online news reading," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 435–444, 2018.

[23] M. Clements, A. P. d. Vries, and M. J. Reinders, "Exploiting positive and negative graded relevance assessments for content recommendation," in *International Workshop on Algorithms and Models for the Web-Graph*, pp. 155–166, Springer, 2009.

[24] Y.-C. Chen, Y.-S. Lin, Y.-C. Shen, and S.-D. Lin, "A modified random walk framework for handling negative ratings and generating explanations," *ACM transactions on Intelligent Systems and technology (tISt)*, vol. 4, no. 1, pp. 1–21, 2013.

[25] D. H. Lee and P. Brusilovsky, "Reinforcing recommendation using implicit negative feedback," in *International conference on user modeling, adaptation, and personalization*, pp. 422–427, Springer, 2009.

[26] Y.-L. Chen, Y.-H. Yeh, and M.-R. Ma, "A movie recommendation method based on users' positive and negative profiles," *Information Processing & Management*, vol. 58, no. 3, p. 102531, 2021.

[27] A. Ferraro, D. Bogdanov, and X. Serra, "Skip prediction using boosting trees based on acoustic features of tracks in sessions," *arXiv preprint arXiv:1903.11833*, 2019.

[28] N. Montecchio, P. Roy, and F. Pachet, "The skipping behavior of users of music streaming services and its relation to musical structure," *Plos one*, vol. 15, no. 9, p. e0239418, 2020.

[29] S. Chang, S. Lee, and K. Lee, "Sequential skip prediction with few-shot in streamed music contents," *arXiv preprint arXiv:1901.08203*, 2019.

[30] J. Pons and X. Serra, "musicnn: Pre-trained convolutional neural networks for music audio tagging," *arXiv preprint arXiv:1909.06654*, 2019.

[31] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *2017 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pp. 2392–2396, IEEE, 2017.

[32] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, "Evaluation of algorithms using games: The case of music tagging.," in *ISMIR*, pp. 387–392, 2009.

[33] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," 2011.

[34] A. Van Den Oord, O. Vinyals, *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[35] A. Razavi, A. Van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with vq-vae-2," *Advances in neural information processing systems*, vol. 32, 2019.

[36] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The mtg-jamendo dataset for automatic music tagging," 2019.

[37] G. Jawaheer, M. Szomszor, and P. Kostkova, "Comparison of implicit and explicit feedback from an online music recommendation service," in *proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems*, pp. 47–51, 2010.

[38] I. Peretz, D. Gaudreau, and A.-M. Bonnel, "Exposure effects on music preference and recognition," *Memory & cognition*, vol. 26, no. 5, pp. 884–902, 1998.

[39] A. Bellogin, P. Castells, and I. Cantador, "Precision-oriented evaluation of recommender systems: an algorithmic comparison," in *Proceedings of the fifth ACM conference on Recommender systems*, pp. 333–336, 2011.

[40] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.

[41] G. Shani and A. Gunawardana, "Evaluating recommendation systems," in *Recommender systems handbook*, pp. 257–297, Springer, 2011.

[42] E. Mena-Maldonado, R. Cañamares, P. Castells, Y. Ren, and M. Sanderson, "Agreement and disagreement between true and false-positive metrics in recommender systems evaluation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 841–850, 2020.

[43] P. Sánchez and A. Bellogín, "Measuring anti-relevance: a study on when recommendation algorithms produce bad suggestions," in *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 367–371, 2018.

[44] D. Kotkov, S. Wang, and J. Veijalainen, "A survey of serendipity in recommender systems," *Knowledge-Based Systems*, vol. 111, pp. 180–192, 2016.

[45] T. Schäfer and P. Sedlmeier, "From the functions of music to music preference," *Psychology of Music*, vol. 37, no. 3, pp. 279–300, 2009.

[46] P. J. Rentfrow, L. R. Goldberg, and D. J. Levitin, "The structure of musical preferences: a five-factor model.," *Journal of personality and social psychology*, vol. 100, no. 6, p. 1139, 2011.

[47] T. Schäfer and P. Sedlmeier, "What makes us like music? determinants of music preference.," *Psychology of Aesthetics, Creativity, and the Arts*, vol. 4, no. 4, p. 223, 2010.

초 록

머신러닝의 발전과 함께 이를 활용한 다양한 음악 추천 시스템이 도입되고 있다. 그러나 음악 추천 시스템에 대한 사용자의 만족도를 높이기 위해서는 단순히 복잡하고 성능이 좋은 모델을 적용하는 것이 아닌, 사용자의 음악 취향에 대한 이해가 반영된 음악 추천 시스템을 설계해야 한다. 비선호도를 활용한 음악 추천 시스템 역시 여러 연구에서 제안되었는데, 비선호도를 반영함으로써 성능이 향상됨을 보였지만 비선호도를 반영하는 것이 구체적으로 어떻게 더 나은 추천으로 이어졌는지에 대한 설명은 부족했다.

본 연구를 통해 우리는 선호도와 비선호도를 다르게 적용하여 훈련된 대조 학습 모델(Contrastive Learning Exploiting Preference, CLEP)을 비교 분석함으로써 사용자의 음악 취향에서 비선호도가 어떤 역할을 가지는지에 대해 알아보고자 한다. 본 연구에서 소개하는 모델은 반영하고자 하는 선호도에 따라 다르게 학습되는 세 가지 모델을 선호도와 비선호도를 모두 반영한 모델(CLEP-PN), 선호도만을 반영한 모델(CLEP-P), 비선호도만을 반영한 모델(CLEP-N)로 나뉜다.

본 연구에서 제안한 각 모델의 훈련 및 평가를 위해서 설문조사를 통해 개인 선호도가 포함된 소량의 데이터셋을 구축하였다. 구축한 데이터셋에 대해 각 모델들의 평가 결과를 비교하여 음악 취향에서의 비선호도의 특징과 음악 추천 시스템에서 비선호도를 활용할 수 있는 가능성에 대해 추가로 조명한다. 또한, 음악 데이터로부

터 특징을 추출하는 과정에서 사전 학습된 서로 다른 세 가지 모델을 이용하였으며, 특징 추출기와 무관하게 일관된 경향성의 결과를 보여 제안 방법의 안정성을 입증하였다.

주요어: 콘텐츠 기반 음악 추천, 비선호도, 대조학습
학 번: 2021-24997