Method to Control Embedded Representation of Piece of Music in Playlists

Hiroki Shibata(✉), Kenta Ebine, and Yasufumi Takama

Graduate School of Tokyo Metropolitan University, Hachioji, Japan
{hshibata,ytakama}@tmu.ac.jp, kenta-ebine@ed.tmu.ac.jp

Abstract. This paper proposes a new method to control embedded representation of a piece of music provided as a set of playlists. To recommend an appropriate piece to a user, numerical representation of music has been studied so far. This study does not focus on explicit representation like signal data, rather implicit representation called embeddings obtained from users' playlists in order to avoid issues around copyright. In the previous work, naive method was proposed and raw dataset is provided to learn the model of embedding for pieces of music, however, it is still not clear the raw dataset is appropriate for the model of music recommender system. Actually, this paper shows there is bias in a raw dataset and it makes the representation tends to provide trivial result, i.e., clearly same element that can be implied only by shallow knowledge like that pieces of music composed by the same artist are similar each other. This study shows the problem quantitatively and proposes a new method to reduce self-evident feature from embedded representation.

Keywords: Music Recommendation · Learning Representation · Distributed Representation

1 Introduction

Nowadays, music streaming services such as Spotify[1] and Apple Music[2] have enabled people to listen to vast amount of pieces of music. In this situation, demand to music recommender system to provide appropriate contents from the sea of music is increasing. Music reommender system is expected to help people find what they most prefer with practical effort, without checking entire data. In addition, not only taking each piece of music one by one, playlist has been getting users' attention as the way to listen to music. Playlists are also used to share the favorite pieces of music among users. Spotify mentioned above actually provide a service to create and share the playlist. Soundcloud[3] is another services that provide the way to share playlists by the users.

[1] https://www.spotify.com/jp/.
[2] https://www.apple.com/jp/apple-music/.
[3] https://soundcloud.com.

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2024
B. Xin et al. (Eds.): IWACIII 2023, CCIS 1931, pp. 226–240, 2024.
https://doi.org/10.1007/978-981-99-7590-7_19

To recommend an appropriate piece of music to a user by computing, numerical representation of a piece of music has been studied so far, such as explicit representation like signal data, and statistical data of the signal. However, this study does not focus on such explicit data in order to avoid issues arround copyright. Instead, the study focuses on implicit representation (numerical representation) called embeddings obtained from users' playlists. As the embedding, this paper focuses on distributed representation that is often employed in a domain of Natural Language Processing (NLP) [1–3]. Distributed representation is applied to various tasks such as, obtaining a embedding vector for each item in a recommendation system [4,5], link prediction in social networking services [6]. It is also applied to music recommendation by introducing embedding representation of pieces of music [13] with those methods to obtain distributed representation in NLP.

Core functionality of the distributed representation is making a vector so called embedding representation of words in a corpus that represents their words. It is archived based on the distributional hypothesis that means "the meaning of a word is generated by its neighbor words [1]." The original hypothesis is only applied to a text, however, it can be applied to any sequence of finite variety of elements if there is a rule among their arrangement similar to the syntax that defines a sentence of natural language. Actually there are studies to propose generating embeddings of pieces of music from the listening history of users [13], in which music titles are lined up in temporal axis. However, the authors suppose its investigation of obtained embedding representation is still insufficient (problem 1), and there was a problem of biases by which embeddings represent trivial feature of the dataset, in which, most pieces of music in high raked position that are placed close to a certain piece in the embedding space belong to the same artist as that of the piece (problem 2). It lacks serendipity. As for expected functionality in practical use, users want to be recommended pieces (here Y), that are similar to a piece (here x) a target user likes but not from the same artists as who composed x, because the user probably knows Y already.

To these 2 problems (problem 1, 2) above, the contributions of this paper are followings.

- Providing a quantitative investigation for the obtained embeddings
- Proposal of a new method to control the embeddings to reduce trivial feature, similar pieces of music are almost from the same artists

From these results above, the paper contributes further development of a method to obtain a meaningful embeddings and gives detailed insight to embeddings of piece of music from music playlist, towards practical implementation of music recommender system.

1.1 Notations

Notations that are supposed to uncommon is explained here. Although all notations are explained at their emerging point too, please see the section for convenience.

For positive integer is denoted as $\mathbb{N}$. For $n \in \mathbb{N}$, $[n] = \{1, 2, ..., n\}$, set of all integers from 1 to n. $\delta(a, b)$ is a Kronecker's delta that takes 1 if $a = b$ and 0 otherwise. $\chi(x; A)$ is an index function that takes 1 if $x \in A$ and 0 otherwise. $x \in A$ means x is belongs to A, this definition related to a set is follows set theory. For 2 set A, B, $A \cap B$, $A \cup B$, $A - B$ are intersection, union, and difference of A and B. $f : A \to B$ denotes a mapping, noted also as $f(a) = b$ with some example of $a \in A, b \in B$.

For given elements $a_1, a_2, ..., a_n$, a finite set A consists of these elements is denoted as $A = \{a_1, a_2, ..., a_n\}$, and sequence (or series) of these elements is denoted as $(a_1, a_2, ..., a_n)$ or $(a_i; i \in [n])$. That is, $\{\}$ is used for a set, $()$ is used for a sequence that the order is defined by its index and multiple occurrences of the same element is possible. A^n is a set of all tuples (or sequence, or pairs in the case of 2) of $a = (a_i; i \in [n])$ (Note that pairs, tuples, sequence and series are all the same concept.). $\#A$ is a number of elements in a set A.

2 Related Work

2.1 Distributed Representation

The distributed representation of word is an embedding of words in its vocabulary on a vector space (typically, numerical vector space with the dimension $N > 1$) [2,3]. Each vector of a word is calculated (learnt) under the set of restriction, called model of the distributed representation. These vectors for words are called embedding and its space to which the vectors belong is called embedding space frequently. In the model, words have similar meanings are placed closer point to each other, compared with those have distant meanings. Those vectors show additive property, such that if the model predicts $v(w), w \in W$ that is closest to $v(\text{king}) + v(\text{female}) - v(\text{male})$, it will be $v(\text{queen})$, as well known, where $v : W \to R^D$ denotes a mapping from W to a numerical vector space R^D. Here W is a set of words (vocabulary). Therefore, in this model, tasks that need the meaning of words can be done with commonly known algebraic operation of vectors in a vector space, which provide consistent and quantitative way of the task. Typical dimension of the embedding space D is $100 \leq D \leq 1000$ [2].

As the representative model and method of the distributed representation of words, there is Word2Vec [2,3]. It models a mapping from words (W) to a embedding vector space with 2 layers Neural Networks, and it is divided into 2 sub-categories called CBOW (Continuous Bag-of-Words) and Skip-gram. A model based-on CBOW is defined so that it solves a problem to predict a word (here w $(\in W)$) provided that the w's surrounding words are given. Surroundings means a neighbor relation in a sequence. More strictly, it is given as a sequence of words $(w_i; i \in [N]), w_i \in W$, where there is a $i \in [N]$ that holds $w_i = w$, and

for positive integer A, $[A]$ means $[A] = \{1, 2, ..., N\}$. Note that usually $\#W < N$. the task is that for all $i \in [N]$, the model is learnt so that it predicts a word placed at i only with the sequence of words $(w_j; j \in [i + c] - [i - c - 1] - \{i\})$, where c is called the window size. Because there is multiple $i \in I(w') \subset [N]$ that $w_i = w'$ for each $w' \in W$, prediction for each place of $i \in I(w')$ cannot be done exactly, that is, there is a tradeoff among occurrences of w'.

On Skip-gram model, on the other hand, for each place i, the task predicts a word for each place $j \in [i + c] - [i - c - 1] - \{i\}$. Skip-gram is more time complicated CBOW but it usually gives better results than CBOW [2].

2.2 Music Recommendation

There are studies on music recommendation, such as [7–9,11–14]. This study focuses on music playlists and embedding for pieces of music obtained from the playlists, because playlists are everywhere on music streaming services nowadays and does not depend on the contents of music, which enables the system avoid copyright related issues. That is, it can be expected to be available in the future continuously. Moreover, it explains music in groups that express users' subjective categorization of music, which enables the system to understand not only users' preferences to the music, but also the situation that music is possibly listened to. Speaking to applicability, embedding of distributed representation has variety of advantages, therefore, music embedding from playlists is worth studying. This section introduces representative work for embeddings of music from playlists under the assumption of distributed representation.

Wang et al. [13] have proposed a method to calculate a vector for each piece of music by, regarding sequences of pieces listened to by users successively at once, as sentences dealt with in Skip-gram-based Word2Vec. Recommendation is done using user-based collaborative filtering (CF) in which similarity between any pair of users needed in CF is calculated based on the above-mentioned vectors. The main focus of this study [13] is to recommend a piece of music, rather than a playlist of music. In addition, a qualitative discussion on obtained vector is provided in which it is mentioned that vectors among the same artists are similar to each other. However, over all analysis is remained just in showing visualization without quantitative evaluation of results. In addition, there is a problem in the method that most high ranked pieces of music come from the same artists and the results of recommendation lacks serendipity (diversity).

Zhou et al. [14] have proposed a method based on CBOW-based Word2Vec to calculate a vector for each piece of music using all listening history of users. At the recommendation for a user u, 2 vector $\bar{a}, \bar{b}$ is calculated first where $\bar{a}$ is a mean vector of all vector for pieces of music in user u's listening history while $\bar{b}$ is a mean vector for recent listened to pieces of music by user u. Then a piece of music with a vector c that closest to $(\bar{a} + \bar{b})/2$ is recommended to u. This study also focuses on recommending just a piece of music, but does not provide any quantitative analysis for obtained embedding vectors. In addition, the proposed method from [14] does not provide a discussion to remove bias mentioned above either.

There are also studies that proposes to generate and recommend a playlist itself. Claudio et al. [16] have proposed a method to generate a playlist for recommendation by taking a piece of music designated by a user as an input. Flexer et al. [17] have proposed a playlist generation method that recognizing user's context at the recommended time. It employs signal characteristics and lyrics of music. Although its applicability is limited because it uses specific type of data such as lyrics and signal characteristics, it shows practical recommendation method. However, in this paper, we focus on more general formalization by employing only the playlists and seek its capability as the playlists data is ubiquitous as mentioned in the introduction.

3 Proposed Method and Investigation

This paper proposes a new method to control the embedding vectors for pieces of music obtained from playlists as dataset of which sequences are assumed to hold the distributional hypothesis that, "latent structure that characterizes a sequence of playlist exists and the structure can be employed to obtain a effective embedding representation for music recommendation [1]." As a model for embeddings, Word2vec [2] is employed following to Wang et al. [13]. To the best of our knowledge, quantitative investigation to the characteristics of learnt embedding vectors has not been conducted sufficiently, therefore, in addition to the proposal of a new method, the present paper provides a quantitative analysis to the obtained embeddings.

For dataset, Spotify Playlists Dataset[4] that is published by Martin et al. [18], is used to obtain embeddings of pieces of music. This data set consists of playlists published by users who posted tweets on Twitter[5] with hash tag "#nowplaying" via Spotify. Number of composition titles in the dataset is 12,867,130. Each entry of dataset consists of, "user_id" (user name on Spotify), "artist name," "track name," "playlist name." Playlist id is assigned at the preprocessing period combining user_id and playlist name so that each string is identical in the dataset. After that, the dataset has 188,437 playlists. 88% of playlists has less than 100 pieces of titles, while the longest playlists has 47,362 titles. Common parameter configurations are,

$$c = 9, D = 50, N_{\mathrm{s}} = 5,$$

where N_{s} is parameter related to the number of negative samplings used at the learning. Actual implementation of the model relays on Gensim[6] of Python's library.

Here is common definitions in experiments and investigation. As a similarity, here after cosine similarity denoted as $\cos(u, v)$ for vectors u, v is used. Let a set of pieces of music M, Let a set of playlist P, its element be $p = (p_i \in M; i = 1, 2, ...) \in P$, the length of p be n_p.

[4] https://dbis.uibk.ac.at/node/263.
[5] https://twitter.com.
[6] https://radimrehurek.com/gensim/index.html.

Table 1. 10 pieces of music that each $m \in M(m*)$ of them has the 10 highest similarity of $\cos(m_*, m')$ from the top.

Similar songs m' to m_*	$\cos(m_*, m')$
michael jackson - bad	0.925
michael jackson - billie jean	0.923
michael jackson - beds are burning	0.861
michael jackson - beautiful firl	0.856
michael jackson - black or white	0.856
michael jackson - ben	0.844
the jacksons - beat it	0.839
meat loaf - bat out of hell	0.835
the jacksons - blame it on the boogie	0.834
u2-beautiful day	0.829

3.1 Investigation on Embeddings

In distributed representation, embeddings for similar words will be placed near by each other, however, it is not clear that property also holds for the case of dataset on pieces of music sequences (playlists). In this section, it is investigated.

An qualitative investigation about similar pieces of music is provided first. Letting m_* = "michael jackson - beat it", and $M(m*)$ be a set of those $m' \in M$ with top 10 highest $\cos(m_*, m')$, Table 1 shows the list of artist-titles $m' \in M(m_*)$ with corresponding value $\cos(m_*, m')$. From the table, most of artists are the same. That means, estimated with distributed representation of pieces of music tend to be the same artists with the focused piece. This tendency coincides with that reported by Wang at el. [13].

Next the paper provide an investigation into similarity of pairs of pieces occur in the same playlists. Additional mathematical formulations are needed for statistics. So those are defined first. Subset M^γ of M is defined as,

$$M^\gamma = \left\{ m; \sum_{i \in n_p} \delta(p_i, m) \leq 1, p \in P, m \in M \right\},$$

where δ is Kronecker's Delta. The investigation is conducted on this subset M^γ. Let the frequency of co-occurrence be $f : M^2 \to \mathbb{N}$. f is defined for $m, m' \in M$ as,

$$f(m, m') = \sum_{p \in P_m} \sum_{i \in [n_p]} \delta(m', p_i), P_m = \{p; \exists i \in [n_p] \, [p_i = m], p \in P\}.$$

Then defining a sorted sequence $(l_i^m; i \in [\#M^\gamma], l_i^m \in M^\gamma), i \neq j \Rightarrow l_i^m \neq l_j^m$ so that $l_i^m \geq l_{i+1}^m$ with the order $\geq$: $m' \geq m'' \Leftrightarrow f(m, m') \geq f(m, m'')$, letting a set

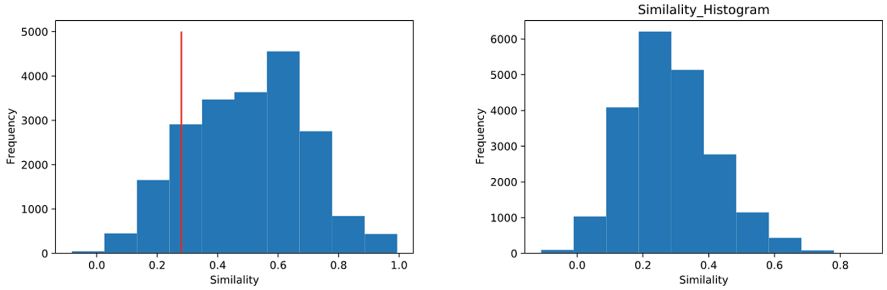

Fig. 1. (a): A histogram of cosine similarity between similar pairs in $\mathfrak{M}$. Vertical line in the middle denotes a mean position of the histogram in (b). (b): A histogram of cosine similarity between un-similar paris in $\mathfrak{M}^*$.

of similar piece to m be S_m defined as $S_m = \{l_i^m; i \in [\alpha]\}$, a set of similar pairs $\mathfrak{M}$ is defined as,

$$\mathfrak{M} = \{(x, x') ; g(x, x') \geq \beta, x, x' \in M^\gamma\}, g(x, x') = \sum_{m \in M^\gamma} \chi(x; S_m)\chi(x'; S_m), \tag{1}$$

where $\chi(x; X) \in [0, 1], x \in X \Leftrightarrow \chi(x; X) = 1$ is an index function, and $\alpha = 30, \beta = 3$ is used in this experiment. Note that the higher $g(x, x')$, the more similar the sets to which x, x' belong. In analogy in word sequence (sentence), if $g(x, x')$ take higher value, this pair (x, x') will occur in the same sentences frequently, that is, these has the similar or related meanings each other. Such words should placed in the near position on embedding space of models based on the assumption of distributed representation [1]. Let non-similar set of pairs of pieces of music be $\mathfrak{M}^* \subset (M^\gamma)^2 - \mathfrak{M}$. $\mathfrak{M}^*$ is defined by collecting pairs $(x, x') \in (M^\gamma)^2 - \mathfrak{M}$ at random.

Histogram of cosine similarity between pairs for each $\mathfrak{M}, \mathfrak{M}^*$ is prepared in Fig. 1(a), Fig. 1(b) respectively. Parameters are $\gamma = 100, \alpha = 30, \beta = 3$. A vertical line in the middle of Fig. 1(a) denotes a mean position of the histogram in Fig. 1(b) to help the comparison. From the figures, it is confirmed that value of the similarity distributes in higher position of the axis for $\mathfrak{M}$, while in lower position for $\mathfrak{M}^*$ than for $\mathfrak{M}$. Average similarities for $\mathfrak{M}, fM^*$ are $0.5, 0.28$ respectively indeed. These results indicate similar piece of music has closer embedding vectors each other than those of un-similar.

In the next investigation about similarity in sequence 2 piece of music, m_1 = "daft punk-get lucky", m_2 = "m83-midnight city" $\in M^\gamma, \gamma = 1000$ is selected. In addition, 2 playlist $p^1, p^2 \in P$ is selected that contain these 2 piece respectively, that is, $\exists i \in [n_{p^j}] \left[m_j = p_i^j\right], j = 1, 2$. Graphs of cosine similarity, $\cos\left(m_j, p_i^j\right), i \in [n_{p^j}]$ are shown for each $j = 1, 2$ in Fig. 2(a), Fig. 2(b) respectively. i corresponds to the horizontal axis named song order in the graph. The position with similarity 1 corresponds to the index i of the

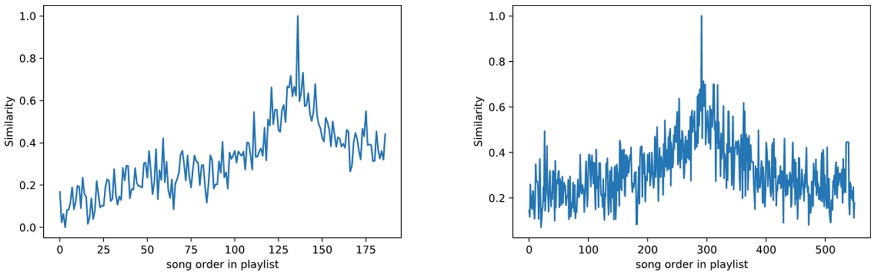

Fig. 2. Inner playlist similarity. (a): $m_1 =$ "daft punk-get lucky" and $p_i^1, i \in [n_{p^1}]$. (b): $m_2 =$ "m83-midnight city" and $p_i^2, i \in [n_{p^2}]$, i corresponds the axis named song order in the graph.

Table 2. Most 5 similar pairs of pieces (m, m'), and cosine similarity $\cos(m, m')$

similar pair(m, m')	$\cos(m, m')$
christophe beck – the north mountain christophe beck – wolves	0.727
christophe beck – sorcery christophe beck – the trolls	0.851
jack white - entitlement jack white – want and able	0.611
childish gambino – playing around before the party start childish gambino – death by numbers	0.774
red hot chili peppers – behind the sun red hot chili peppers – knock me down	0.452

pieces $m_1 = p_i^1, m_2 = p_i^2$. From these figures, piece p_k^j close to m_j has higher similarity than the distant piece p_k^j.

In the next, over all tendency about similarity is investigated. Picking up a set of 100 playlist (E), in which each $p \in E$ holds $n_p \geq 5, p \in P$ at random, the following average similarity μ_d with distance $d = 1, 2, ..., 49$,

$$\mu_d = \frac{1}{\sum_{p \in E} \#I(d,p)} \sum_{p \in E} \sum_{i \in I(d,p)} \cos(p_d, p_{i+d}), I(d,p) = \{i; i \in [n_p], i + d \leq n_p\}$$

is calculated, where $\#X$ for a set X is the number of elements of X. Fig. 3 visualize the relation between the distance d at the horizontal axis and μ_d at the vertical axis. It can be confirmed that until the distance d is less than 20, μ_d is decreasing along with d. The reason in the region $d > 20$, no decreasing tendency of μ_d along with d, can be considered as that this experiment set the windows size of the Word2Vec to 9 that less than 20. In this case, it is supposed that occurrence of pairs with distance larger than 9 within the same window

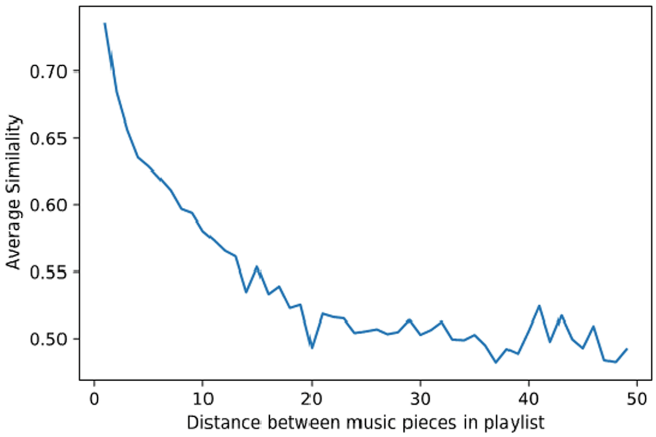

Fig. 3. Relation of distance between pieces of music in a playlist $d = 1, 2, ..., 49$, and average similarity μ_d.

become rare. Lower bound of μ_d seems to be around 0.5. This is higher than 0.28 of average similarity in Fig. 1(b). In this result, a pair in the same window or same playlist has high similarity between each other of the pair.

Those results so far are the intended features of distributed representation. So it is confirmed that models of distributed representation for words can be applied to the sequence of pieces of music.

To see in more detail, Table 2 is prepared. It shows most 5 pairs (m, m') sorted with descending order of the value $g(m, m')$ in Eq. (1). From this table, all the pairs of elements consist the same artist. The reason of this result is considered to be that many users include pieces of music with the same artists within the same playlist. From the results, there is strong tendency that pieces with the same artists have high similarity. Because this is trivial results, the tendency must be changed any time it is needed. To mitigate this problem and change the tendency, this paper proposes a new method to control the characteristics of embedding with respect to their similarity among each other.

3.2 Proposed Method to Reduce Bias

To reduce the chance the same artists appear, the proposed method removes the redundant playlists from the dataset, i.e., playlists that are similar to another with respect to the kind of artists with in them are all removed except one of them. The similarity to measure this redundancy is introduced as following. Letting P be a set of playlists, A be a set of artists in dataset, and $f_p : A \to \in [0, \infty), p \in P, a \in A$ be the realized frequency of a in playlists p, the measure is the following modified Simpson coefficient $t^2 : P \times P \to \mathbb{R}$ defined as,

$$t^2(p, q) = \frac{\sum_{a \in A} \min \{f_p(a), f_q(a)\}}{\min \left\{\sum_{a \in A} f_p(a), \sum_{a \in A} f_q(a)\right\}}. \tag{2}$$

Table 3. Top 10 similar pieces to $m =$ "michael jackson - thriller" with original dataset (W).

artist name and title of piece	similarity
michael jackson - they don't care about us	0.91
michael jackson - this is it	0.88
michael jackson - this time around	0.87
michael jackson - threatened	0.87
michael jackson - the way you make me feel	0.85
cyndi lauper - time after time	0.83
michael jackson - this place hotel	0.81
ke$ha - tik tok	0.80
donna summer - this time i know it's for real	0.79
michael jackson - todo mi amor eres tu	0.79

If $\forall a \in A\,[f_p(a) \leq 1, f_q(a) \leq 1]$, and letting $A_{p'} = \{a; f_p(a) = 1\}$ for $p' \in P$, then t^2 become the following ordinal simpson coefficient t^1,

$$t^1(p,q) = \frac{\#\left(A_p \cap A_q\right)}{\min\left\{\#A_p, \#A_q\right\}}. \tag{3}$$

The procedure of the proposed method is as follows.

$$\hat{P} = \left\{ p; p \in P, \exists a' \in A \left[\sum_{i \in n_p} \delta(a(p_i), a') \geq \frac{n_p}{2} \right] \right\} \tag{4}$$

$$Q_p = \left\{ p'; t^k(p', p) \geq 0.5, p \in \hat{P} \right\}, \hat{Q} = \left\{ \underset{p' \in Q_p}{\operatorname{argmax}} \left(n_{p'}\right); p \in \hat{P} \right\} \tag{5}$$

Then, the proposed method takes $P^* = \hat{Q} \cup (P - \hat{P})$ as the dataset of playlists. It is assumed $\operatorname{argmax}_a f(a)$ represents identical element, that is, its result never change. To the best of our knowledge, there is no study to try to overcome the problem of the existence of the same artists in the nearby vectors by controlling the embedding with modified dataset. Hereafter, results using original dataset P and modified dataset P^* with t^1, and P^* with t^2 are investigated. For simplicity, these results are designated with W, W1, W2 respectively in the paper.

In Table 3, 4, 5, pieces m' similar to $m =$ "michael jackson - thriller" is listed with corresponding similarity $\cos(m, m')$ between embedding vectors m, m'. From these tables, when comparing W1 with W, it can be found that the same artist (michael jackson) in W1 becomes less than half of W. At the same time, although W2 is only different in 3 with respect to the number of the same artists, similarities are decreased from the pieces in W. Because there was not so significant difference in the results W1 and W2, only the results of W and W1 are investigated in the following part.

Table 4. Top 10 similar pieces to m = "michael jackson - thriller" with modified dataset P^* using original simpson coefficient t^1 (W1).

artist name and title of piece	similarity
michael jackson - this is it	0.85
ke$ha - tik tok	0.85
michael jackson – they don't care about us	0.84
cyndi lauper - time after time	0.84
the rock masters - thunderstruck	0.84
richard o'brien - time warp	0.83
donna summer – this time i know it's for real	0.83
maroon 5 - this love	0.82
timbaland - throw it on me	0.81
michael jackson – the way you make me feel	0.80

Table 5. Top 10 similar pieces to m = "michael jackson - thriller" with modified dataset P^* using modified simpson coefficient t^2 (W2).

artist name and title of piece	similarity
michael jackson – they don't care about us	0.89
michael jackson - threatened	0.83
michael jackson – this time around	0.82
michael jackson – the way you make me feel	0.82
maroon 5 - this love	0.82
eurythmics – thorn in my side	0.82
cyndi lauper – time after time	0.82
michael jackson - this is it	0.82
ke$ha - tik tok	0.81
the jacksons – this place hotel	0.81

In Table 6, 7, pieces m' similar to m = "madonna-like a virgin" is listed with corresponding similarity $\cos(m, m')$. The same tendency as the case in michael jackson is observed. Moreover, the piece "david bowie" listed in the high position in the result of W1 is similar with respect to the fact that they are both singer song writer. Therefore, while decreasing similarity among the same artists, it may increase serendipity. Therefore it may suitable to the practical recommendation.

Table 6. Top 10 similar pieces to $m =$ "madonna-like a virgin" with original dataset (W).

artist name and title of piece	similarity
madonna - like a prayer	0.98
madonna - like it or not	0.87
madonna - live to tell	0.85
madonna - la isla bonita	0.82
deniece williams - let's hear it for the boy	0.84
bon jovi - livin' on a prayer	0.84
roxette - listen to your heart	0.82
madonna - like a prayer 2008	0.82
madonna - like a virgin/hollywood	0.81
fleetwood mac - little lies	0.81

Table 7. Top 10 similar pieces to $m =$ "madonna-like a virgin" with modified dataset P^* using modified simpson coefficient t^1 (W1).

artist name and title of piece	similarity
madonna - like a prayer	0.98
roxette - listen to your heart	0.89
deniece williams - let's hear it for the boy	0.83
fleetwood mac - little lies	0.83
david bowie - let's dance	0.83
bon jovi - livin' on a prayer	0.82
madonna - la isla bonita	0.82
thompson twins - lies	0.81
madonna - like it or not	0.81
prince - little red corvette	0.81

The results for $m =$ "james blunt-i'll be your man" is shown in Table 8, 9. For these results, significantly, the number of the same artists decreased from W to W1. Its ranking of the original artist (james blunt) decreased as well. It is supposed that the proposed method has bigger effect to decrease the similarity of unfamiliar artists.

Table 8. Top 10 similar pieces to m = "james blunt-i'll be your man" with original dataset (W).

artist name and title of piece	similarity
james blunt - i'll take everything	0.87
james blunt - if time is all i have	0.87
james morrison - i won't let you go	0.81
jason mraz - i won't give up	0.79
james blunt - i really want you	0.79
krezip - i would stay	0.79
jason mraz - i'm yours	0.79
lenny kravitz - i'll be waiting	0.78
noel gallagher's high flying birds – if i had a gun	0.78
joshua radin - i'd rather be with you	0.78

Table 9. Top 10 similar pieces to m = "james blunt-i'll be your man" with modified dataset P^* using modified simpson coefficient t^1 (W1).

artist name and title of piece	similarity
olly murs - i've tried everything	0.85
james morrison - i won't let you go	0.83
gavin mikhail – i will follow you into the dark	0.83
lenny kravitz - i'll be waiting	0.82
jason mraz - i'm yours	0.82
joshua radin - i'd rather be with you	0.82
jason mraz - i won't give up	0.81
owl city - i'll meet you there	0.81
james blunt - if time is all i have	0.79
james arthur - impossible	0.79

4 Conclusion

This paper shows quantitative analysis of embedding vectors for a piece of music obtained with the model based on the assumption of distributed representation, with the implementation on Word2Vec. The intended feature of embedding vectors was observed, in which pieces of music are placed in closer position with the similar pieces and distant position from the un-similar pieces on the embedded vector space, where the similarity among pieces of music is defined by the position in the playlist. It was confirmed the analogy replacing a concept of sentence in the original Word2Vec with a playlist of pieces of music is valid from quantitative investigation in the paper. Furthermore, this paper discussed the problem especially exists in employing the raw playlist as a training dataset,

i.e., most of the pieces of music close to each other in the embedding space come from the same artist. To overcome the problem, the paper proposed a method to control generated embedding vector by manipulating the dataset, which reduce the redundant playlists that consist of almost one artist. The way of manipulation was intuitive and easy to understand. Moreover, the results was consistent with the expected effect, and shows suitable characteristics when it is considered to be applied in practical recommendation, as it increases the serendipity of recommended list of pieces.

As the remaining work, investigation must be done more concretely. For the aim, selection of suitable measure to evaluate the statistics for music recommendation is needed. By applying proposed method as the component of a recommender system, it is expected that recommendation with high user's satisfaction and attention can be archived.

Acknowledgements. This work was partially supported by JSPS KAKENHI Grant Numbers 21H03553, 22H03698, and 22K19836.

References

1. Harris, Z.S.: Distributional Structure. WORD **10**(2–3), 146–162 (1954)
2. Mikolov, T., Chen, K., Corrad, o G., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of International Conference on Learning Representations Workshops Track (2013)
3. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. Adv. Neural. Inf. Process. Syst. **26**, 3111–3119 (2013)
4. Krishnamurthy, B., Puri, N., Goel, R.: Learning vector-space representations of items for recommendations using word embedding models. Procedia Comput. Sci. **80**, 2205–2210 (2016)
5. Yoon, Y., Lee, J.: Movie recommendation using metadata based Word2Vec algorithm. In: 2018 International Conference on Platform Technology and Service, pp. 1–6 (2018)
6. Amiri, M., Shobi, A.: A link prediction strategy for personalized tweet recommendation through Doc2Vec approach. Res. Econ. Manag. **2**(4), 63–76 (2017)
7. Baltrunas, L., et al.: InCarMusic: context-aware music recommendations in a car. In: Huemer, C., Setzer, T. (eds.) EC-Web 2011. LNBIP, vol. 85, pp. 89–100. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23014-1_8
8. Yapriady, B., Uitdenbogerd A.: Combining demographic data with collaborative filtering for automatic music recommendation. In: 9th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, pp. 201–207 (2005)
9. Radhika, N.: Music recommendation system based on user's sentiment. Int. J. Sci. Res. 383–384 (2015)
10. Jawaheer, G., Szomszor, M., Kostkova, P.: Comparison of implicit and explicit feedback from an online music recommendation service. In: International Workshop on Information Heterogeneity and Fusion in Recommender Systems (2010)
11. Bogdanov, D., Haro, M., Fuhrmann, F., Gómez, E., Herrera, P.: Content-based music recommendation based on user preference examples. In: Workshop on Music Recommendation and Discovery, Colocated with ACM RecSys 2010 (2010)

12. Cano, P., Koppenberger, M., Wack, N.: Content-based music audio recommendation. In: Proceedings of the 13th ACM International Conference on Multimedia (2005)
13. Wang, D., Deng, S., Xu, G.: Sequence-based context-aware music recommendation. Inf. Retrieval J. **21**, 230–252 (2018)
14. Zhou, Y., Tian, P.: Context-aware music recommendation based on Word2Vec. In: International Computer Science and Applications Conference, pp. 50–54 (2019)
15. Köse, B., Eken, S., Sayar, A.: Playlist generation via vector representation of songs. In: Angelov, P., Manolopoulos, Y., Iliadis, L., Roy, A., Vellasco, M. (eds.) INNS 2016. AISC, vol. 529, pp. 179–185. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-47898-2_19
16. Baccigalupo, C., Plaza, E.: Case-Based Sequential Ordering of Songs for Playlist Recommendation, pp. 286–300. Advances in Case-Based Reasoning, European conference (2006)
17. Flexer, A., Schnitzer, D., Gasser, M., Widmer, G.: Playlist generation using start and end songs. In: International Conference on Music Information Retrieval, pp. 173–178 (2008)
18. Pichl, M., Zangerle, E., Specht, G.: Towards a context-aware music recommendation approach: what is hidden in the playlist name?. In: Proceedings of 15th IEEE International Conference on Data Mining Workshops, pp. 1360–1365 (2015)