Vol.2019-MUS-123 No.21
Vol.2019-SLP-127 No.21
2019/6/22

度数表記と Chord2Vec を利用した楽曲類似度指標の提案

石田颯人[†1]　木村昌臣[†2]

概要: 作曲において, ある楽曲を参考にする際, 当該楽曲と類似した楽曲を探すのは困難である. そこで, 本研究では, 音響音楽信号のみを用いた楽曲間の類似度に関する指標を提案する. まず, 音響音楽信号から和音および調性を推定し, 度数表記された和音列へと変換する. 次に, 得られた和音列に Word Embedding を応用することで(Chord2Vec), 和音の分散表現を獲得する. 最後に, 和音列の前後関係を考慮するために拡張した N-gram を導入することで, 楽曲間の類似度を算出する. 本研究の評価として, 提案手法の類似度と被験者が感じた類似度の比較を行った.

キーワード: word2vec, Chord2Vec, N-gram, 楽曲類似度指標, カデンツ, 度数表記, クロマベクトル, クロマグラム

1. はじめに

近年、音楽配信の発展に伴い, 楽曲が音響音楽信号として楽曲データベースに蓄積されている. 楽曲データベースからユーザへ楽曲を推薦する方法として, 楽曲のジャンルや作曲者, 歌手といった情報に基づいた推薦が行われている. しかし, 音響音楽信号以外の情報を扱った推薦では, 楽曲そのものが似ている楽曲を推薦することは難しい.

モーツァルトやショパンがバッハの楽曲から影響を受けたとされるように音楽家は既に存在する音楽から影響を受ける. モーツァルトはバッハのような楽曲を求め, ヘンデルの楽曲も聴取したとされる. どちらの楽曲にも作曲技法の一つである対位法が用いられた楽曲として分析されており, このことから類似している楽曲を推薦するための指標は楽曲の分析によって明らかになる可能性が示唆されているといえる.

本研究は, 和音に焦点を当てた楽曲類似度算出手法を提案することで, 類似楽曲の推薦への貢献を目的とする.

ピッチクラスという 12 種類の音の構成に基づいて和音の名前が定まることから, 任意の時間において発音されているピッチクラスの割合を考慮する必要がある. また, 一般に, 任意の和音は調整に基づいて機能が定まるとされていることから調整に基づいた和音の機能を考慮する. さらに, 音楽は時間に基づいて様々な音が連なることから, 和音の遷移を考慮する必要がある. 以上のことから, 本研究では, 「ピッチクラスの割合」, 「調整に基づいた和音の機能」, 「和音の遷移」が考慮された類似度を楽曲類似度と定義し, 高い楽曲類似度を持つ楽曲同士を類似楽曲と定義する.

2. 前提となる知識

我々は空気振動を知覚した際, その振動の周波数の高さから音の高さを知覚することができる. 2 つの音の高さの差を音程と呼び, 特に周波数が 1:2 の関係にあるような音程はオクターブの関係にあると呼ばれ, .オクターブの関係にある 2 種類の音は同一の音であると知覚される. また、一般に音楽は 440Hz 付近の周波数を基準とし, これを基準周波数と呼ぶ. 基準周波数からそのオクターブの関係にある周波数までを均一に 12 等分した各周波数及びそのオクターブの関係にある周波数に対してそれぞれピッチクラスという 12 種類の音の名前が定義される.

一般に, ピッチクラスはアルファベット 7 つと臨時記号を用いて表現される. 臨時記号はピッチクラスとともに用いられ, 当該ピッチクラスの半音上を表す ♯(シャープ)と当該ピッチクラスの半音下を表す♭(フラット), 当該ピッチクラス自身を表す♮(ナチュラル)がある. アルファベットと臨時記号を用いたピッチクラスの例を五線譜とともに図 1 に示す.

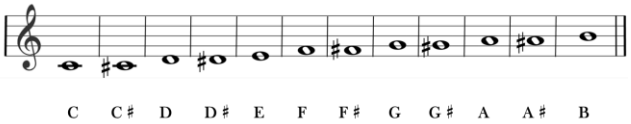

図 1 アルファベットと臨時記号によるピッチクラスの例

また, 人間は同時刻に鳴っている複数の音高を知覚することができる. 音楽では, 同時に知覚された複数の音高のまとまりを和音と呼び, このとき和音の中心になる音を根音という. また, 和音は構成される複数の音の音程関係から命名される. 例えば, C, E, G というピッチクラスからなる和音は C メジャー(C major)と命名され, しばしば C や C△と表記され, D, F, A というピッチクラスからなる和音は D マイナー(D minor)と命名され, しばしば Dm と表記される. 本研究ではピッチクラスとしての C と和音の名称(以下, 和音名) としての C の混合を避けるため, C メジャーを C△のように表記する.

任意の楽曲において知覚的に中心となるピッチクラスを主音と呼び, 主音とその他のピッチクラスとの位置関係からなる体系を調性という. また, 主音を根音として構成

†1 芝浦工業大学大学院
Graduate School, Shibaura Institute of Technology
†2 芝浦工業大学
Shibaura Institute of Technology

Vol.2019-MUS-123 No.21
Vol.2019-SLP-127 No.21
2019/6/22

された和音を主和音と呼び, 最も安定した印象を知覚させる機能をもった和音として解釈される. 和音の機能について, 主和音以外の和音の機能は主和音との相対的な音程関係から定まるとされる[1]. 主音とその他のピッチクラスとの位置関係によって調性が定まり, 主和音との相対的な音程関係によって和音の機能が定まることから、任意の和音の機能は調性の考慮なしに一意に定まらない. 例えばピッチクラス C を主音とする楽曲において和音 G△はしばしば緊張感を知覚させる機能をもつが, ピッチクラス G を主音とする楽曲において和音 G△は主和音であるから最も安定した印象を知覚させる機能をもつ. このことから, 任意の和音は調性によって機能が異なること, すなわち任意の和音は多義性をもつことがいえる.

そこで, 調性に基づかず, 主和音との相対的位置のみに注目した和音の表記方法として度数表記が多用される[1]. 度数表記とは, 主音と各和音の根音との相対位置をギリシャ数字と臨時記号で表記する方法である, また, 度数表記は和音にも用いられ, 当該和音のピッチクラスの構成に基づいて和音の命名規則をもとにギリシャ数字と臨時記号で表記される. 度数表記によって主和音との相対位置を明確に表記することで和音の機能は調性によらず一意に定まる, すなわち度数表記された和音は多義性を持たない. 例えば度数表記において, V △は緊張感を知覚させる機能を持つ和音であり, I△は最も安定した印象を知覚させる和音である. ピッチクラスによる和音及び度数表記による和音の例を C を基音としてメジャー, マイナーについてそれぞれ図 2 に示す.

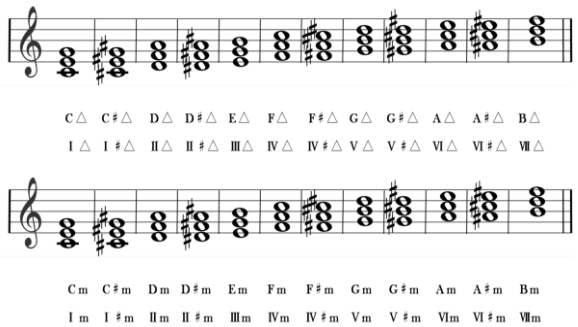

図 2 ピッチクラスによる和音と度数表記による和音

また, 音楽はいくつかの音が時間的に展開されることで構成されるため, これに伴い, 和音も時間的に展開される. 任意の時刻において現在の時刻の和音から次の時刻の和音へ遷移することをカデンツと呼び, カデンツの知覚によってもたらされる期待から, 和音及びカデンツの機能が規定されている.

3. 関連研究

3.1 クロマベクトル

本研究に関連する研究として, 音響音楽信号からコード推定を行う研究がある[2]. この研究では, 音響音楽信号からピッチクラスに基づいたベクトルを抽出する手法として, クロマベクトルという手法を用いている. クロマベクトルまたは Pitch Class Profile とは, 12 種類のピッチクラスについてそれぞれのオクターブの関係にある周波数の信号の強さを足し合わせることで, 任意の時刻における各ピッチクラスの信号の強さの割合が 12 次元で表現されたベクトルである. クロマベクトルは和音の推定のほか, 楽曲の構造分析や音声照合など, 音響音楽信号を用いた研究で頻繁に用いられる

(1) クロマベクトルの抽出

一般的なクロマベクトルの抽出方法の流れを説明する. まず, 入力された音響音楽信号に対し, フーリエ変換及び任意の範囲でのフィルタリングを行った後, ピーク検出を行い, 信号の極大値を取得する. 次に, 基準周波数に基づいて, 周波数値に対する 12 種類のピッチクラスのマッピングを行うことで任意の周波数における信号の強さから各ピッチクラスの信号の強さが推定される. 最後に, 信号の強さに基づいた正規化を行うことでクロマベクトルが得られる.

(2) ディープラーニングによるクロマベクトルの推定

前述したクロマベクトルの推定手法の他に, ディープラーニングを用いたクロマベクトルの推定手法がある[3]. この研究では, MIDI(Musical Instrument Digital Interface)という任意の時刻にある音高の音を鳴らすような指示が定められたデータの規格を用いて和音の推定が行われている. 入力した MIDI と MIDI によって発音された電子楽器のスペクトルを組として, 任意の時刻における各ピッチクラスの強さを推定するように CNN(畳み込みニューラルネットワーク)音響モデルの学習を行っている.

3.2 word2vec

単語の分散表現を用いることで文書間の類似度を算出する手法が word2vec である[4][5]. 分散表現とは, 単語の意味を実数値ベクトルで表現することで, 単語間の演算を行うことができる. word2vec は同じ文脈で出現する単語同士は似た意味を持つ傾向にあるという Haris の分布仮説[6]に基づき, 単語の分散表現を学習している. しかし, 単語の出現順番を学習に用いていないため, word2vec で獲得した分散表現は文の語順が考慮されていない. また, 複数の意味を持っているような単語, すなわち多義性をもつ単語に対応することができない.

(1) Skip-gram

word2vec のモデルの 1 つに Mikolov らが提案した Skip-gram モデルがある [7]. Skip-gram は, ある文から任

Vol.2019-MUS-123 No.21
Vol.2019-SLP-127 No.21
2019/6/22

意の単語を選択し，当該単語を入力単語として当該単語の周辺に共起する単語を出力単語として予測するような教師済み学習を行うモデルである．また，この際の入力単語を注目単語と呼び，出力単語を共起単語と呼ぶ．例えば，“Frederic Chopin is a pianist and composer .”という文において任意の単語の周辺語とみなす幅を 5 単語とし，Chopin を注目単語とすると，pianist や composer が共起単語となる．よって Skip-gram モデルではこのような文を学習に用いた場合，Chopin に対して，pianist や composer が共起する確率をいくつかの次元で表し，この条件付き確率を最大化するように学習することで，いくつかの次元で表された単語の分散表現が獲得できる．また，Skip-gram モデルによって得られた単語の分散表現は加法構成性という単語の分散表現の計算が単語の意味の計算に対応する性質をもつことが知られている．例えば，$vv_{king} - v_{man} + v_{woman}$という計算を行うと$v_{queen}$という結果が得られるような性質がある．しかし，既に存在する単語及び当該単語の周辺に共起する単語を学習に用いるため，学習の際に存在しなかった単語すなわち未知語を用いた計算を行うことができないという問題がある．

3.3 N-gram

本研究に関連する研究として N-gram がある．N-gram とはある文字列に含まれる N 文字までの部分文字列のことで，部分文字列の出現回数をカウントすることを N-gram カウントと呼ぶ．また，N=1 のときユニグラム，N=2 のときバイグラム，N=3 のときトリグラムという． 例えば文字列 ”I have a pen”に対するバイグラムは｛“I”， “I have”， “have a”， “a pen”， “pen”｝があり，その出現回数は全て 1 である．このような部分文字列を出現回数に基づいてベクトルとして扱うことで文の類似度を算出することができる．

類似度関数をコサイン類似度とすると，N-gram によって文字列を変換したベクトルをそれぞれ x, y とするとき，コサイン類似度は次のようになる．

$$\cos(x, y) = \frac{x \cdot y}{|x||y|}$$

4. 提案手法

まず，楽曲データベース内の楽曲ファイルを度数表記された和音列へ変換し，変換後の和音列を学習データに word2vec を用いて，和音の分散表現を獲得する．次に，得られた分散表現を用いて楽曲類似度を算出する．

4.1 手法の概要

本研究では，楽曲データベースとして Free Music Archive[8]で提供されている楽曲ファイルを用いる．

まず，楽曲ファイルから任意の時刻における任意の周波数の信号の強さを取得する．得られた信号の強さをスペクトログラムとして図 3 に示す，次に，クロマベクトルという各ピッチクラスの信号の強さの割合を 12 次元で表現したベクトルを作成する．縦軸をピッチクラス，横軸を時刻として，クロマベクトルを時刻に沿って並べたものであるクロマグラムを図 4 に示す．

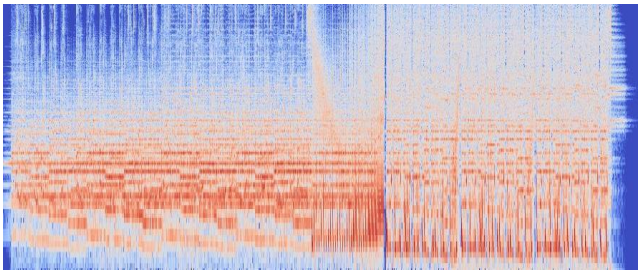

図 3 任意の時刻における任意の周波数の信号の強さを表したスペクトログラム

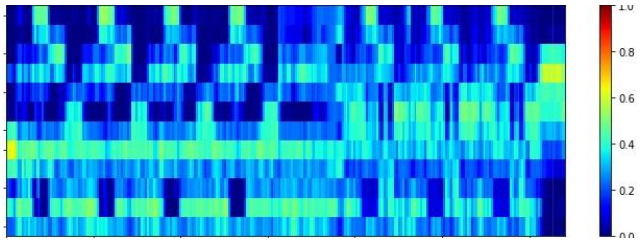

図 4 クロマベクトルを時刻で並べたクロマグラムの例

次に，テンプレートベクトルという，12 次元の和音の特徴を表す量と和音名を対応づけたベクトルを推定する和音名の数だけ用意する．全てのテンプレートベクトルとクロマベクトルの内積を求め，内積が最大であるテンプレートベクトルをクロマベクトルの推定結果として，テンプレートベクトルと対応する和音名を当該クロマベクトルの和音名とする.例えば次のような 12 次元のベクトル[0.5, 0, 0, 0, 0.25, 0, 0, 0.25, 0, 0, 0, 0]を和音名 C と対応づけたテンプレートベクトルと，12 次元のベクトル[0, 0, 0.5, 0, 0, 0, 0.25, 0, 0, 0.25, 0, 0]を和音名 D△と対応づけたテンプレートベクトルの 2 つのテンプレートベクトルがあるとき，クロマグラム[0.4, 0.1, 0, 0, 0.2, 0, 0, 0.2, 0.1, 0, 0, 0]の和音名は C△と推定される(表 1). 作成した全てのクロマベクトルについて和音名の推定を行い，和音名を時刻に沿って順に並べたものを楽曲ファイルの和音列とする．

表 1 クロマベクトルとテンプレートベクトルの比較

ベクトル	
Chroma	[0.4, 0.1, 0, 0, 0.2, 0, 0, 0.2, 0.1, 0, 0, 0]
Vector C△	[0.5, 0, 0, 0, 0.25, 0, 0, 0.25, 0, 0, 0, 0]
Vector Dm	[0, 0, 0.5, 0, 0, 0.25, 0, 0, 0, 0.25, 0, 0]

Vol.2019-MUS-123 No.21
Vol.2019-SLP-127 No.21
2019/6/22

次に，当該和音列の調性の推定を行うことで和音列を度数表記へ変換する．和音推定に用いたテンプレートベクトルと同様に，調性のテンプレートベクトルを作成する．和音列中の和音の出現回数を並べたベクトルと調性のテンプレートベクトルとの内積が最大である和音列を調性の推定結果とする．また，推定した調性に基づき，和音列中のすべての和音を度数表記へ変換する．

次に，和音について word2vec を応用するために，和音の機能とカデンツについて考える．音楽理論において I△と VIm の機能はどちらも安定した印象を与えることから置換可能である．つまり，I△Ⅳ△Ⅴ△というカデンツはⅥmⅣ△Ⅴ△というカデンツに置換可能となっている．このことから，同じ文脈で出現する和音は似た機能を持つことがわかる．すなわち，Haris の分布仮説における単語の分布と和音の分布が対応すると考えられることから word2vec を和音の分散表現の獲得へ応用する．その結果，和音に対するベクトルを得るが，この方法を Chord2Vec と呼ぶこととする．Chord2Vec については，○○[参考文献]による先行研究があるが，度数表記へ変換を行っていない点が本研究と異なる．

ここで，word2vec の多義性の問題があるが，調性に基づいて和音を度数表記に変換することで和音の機能が一意に定まるため，単語の多義性が考慮されないという問題点を無視することができる．また，推定された和音以外の和音は存在しないため未知語の学習にまつわる問題も無視することができる．

4.2 楽曲類似度の算出

楽曲類似度は，算出したい楽曲ファイルを学習データと同様にそれぞれ和音列に変換し，N-gram を拡張した比較を行うことで算出する．

入力する 2 つの楽曲をそれぞれ楽曲 α，楽曲 β とし，度数表記に変換された入力楽曲の和音列に対してそれぞれ，N-gram カウントを行う．次に，楽曲 α と楽曲 β との出現した列に着目し，出現した列の順番に沿って分散表現の比較を行う． 楽曲 α の和音列に対し N-gram カウントを行い，出現した列を出現回数順に並べ，i 番目に多く出現した列内の和音の分散表現を出現順にv_{i1}，v_{i2}，v_{i3}，v_{i4}とし，その出現回数をn_i，出現回数の総和をN_iとする．同様に，楽曲 β の和音列に対し N-gram カウントを行い，出現した列を出現回数順に並べ，j 番目に多く出現した列内の和音を出現順にv_{j1}，v_{j2}，v_{j3}，v_{j4}とし，その出現回数をn_j，出現回数の総和をN_jとする．同様に楽曲類似度は次のように表すことができる．

$$R = \frac{\sum_j \sum_i w_{ij} a_i a_j}{\sqrt{\sum_j \sum_i w_{ij} a_i a_j} \sqrt{\sum_j \sum_i w_{ij} a_i a_j}}$$

ただし，それぞれの記号の定義は以下の通りである．

$$w_{ij} = \sum_k \cos(v_{ik}, v_{jk}), a_i = \sqrt{\frac{n_i^2}{N_i^2}}, a_i = \sqrt{\frac{n_j^2}{N_j^2}}$$

$$\cos(\vec{v}, \vec{u}) = \frac{\sum_i v_i u_i}{\sqrt{\sum_i v_i^2} \sqrt{\sum_i v_j^2}}$$

5. 実験と評価

本研究では，Free Music Archive[数字]で提供されている楽曲ファイルから 100 曲を使用して word2vec モデルを作成した．提案手法で述べた通り，100 曲の楽曲ファイルそれぞれを度数表記列へ変換する操作を行い，100 個の度数表記列を学習データとして word2vec モデルを作成した．

5.1 和音間の類似度の評価

得られたモデルについて，音楽理論において相互に置換可能とされる和音間の類似度を算出したところ，和音Ⅰ△とVIm の類似度が 0.853，Ⅱm とⅣ△の類似度が 0.947，和音Ⅲm と和音Ⅴ△の類似度は 0.680 とそれぞれ高い類似度が得られた．このことから，調性に基づいた和音の機能を保持した分散表現が獲得できていると考えられる．

5.2 楽曲間の類似度の評価

楽曲 A と楽曲 B の類似度を類似度(A, B)，楽曲 C と楽曲 D の類似度を類似度(C, D)と表記する．本研究で提案した手法を用いて，類似度(A, B)，類似度(C, D)をそれぞれ算出した結果を示す(表 2).

表 2 提案手法によって算出した類似度

類似度(A, B)	0.825
類似度(C, D)	0.633

本研究では作曲歴が 3 年以上である被験者を作曲経験者とし，そうでない被験者を作曲未経験者とした．作曲経験者 4 名と作曲未経験者 16 名を対象に楽曲の聴取をしてもらう実験を行った．実験内として楽曲 A と楽曲 B，楽曲 C と楽曲 D のそれぞれの組を聴取する度に，類似度の主観評価を行った．類似度の主観評価方法には VAS 法を用いた．

作曲経験者が感じた類似度(A, B)と類似度(C, D)をそれぞれ示したあと(図 5 図 6)，提案手法による類似度と比較を行う．

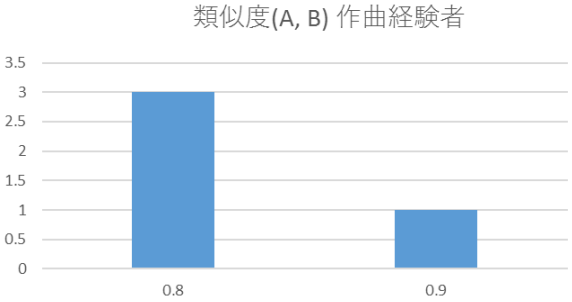

図 5 作曲経験者を対象とした類似度(A, B)の分布

Vol.2019-MUS-123 No.21
Vol.2019-SLP-127 No.21
2019/6/22

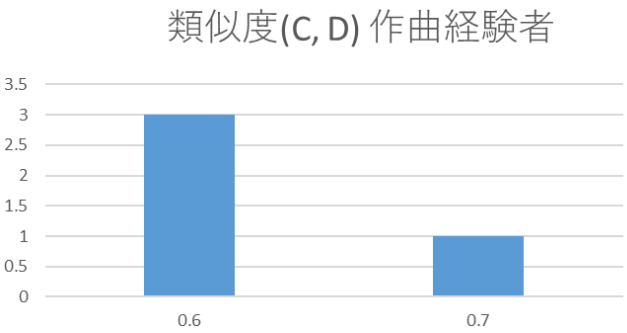

図 6 作曲経験者を対象とした類似度(C, D)の分布

作曲経験者が感じた類似度(A, B)の平均値は 0.825 であり,提案手法によって算出された類似度(A, B)は 0.8.25 である．また，作曲経験者が感じた類似度(C, D)の平均値は 0.625 であり,提案手法によって算出された類似度(C, D)は 0.633 であることから本研究の提案手法による類似度は作曲経験者が感じる類似度に近い類似度が算出できていると考えられる．

次に，作曲経験者が感じた類似度(A, B)と類似度(C, D)をそれぞれ示したあと(図 7 図 8)，提案手法による類似度と比較を行う．

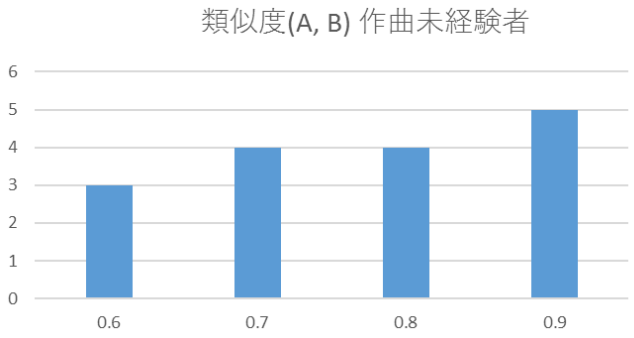

図 7 作曲未経験者を対象とした類似度(A, B)の分布

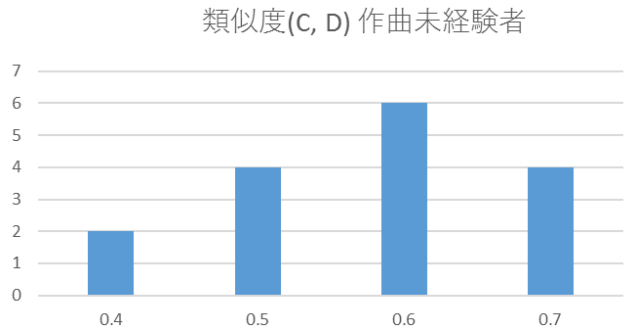

図 8 作曲未経験者を対象とした類似度(A, B)の分布

作曲未経験者が感じた類似度(A, B)の平均値は 0.775 であり,提案手法によって算出された類似度(A, B)は 0.8.25 である．また，作曲未経験者が感じた類似度(C, D)の平均値は 0.550 であり,提案手法によって算出された類似度(C, D)は 0.633 である．

これらの実験結果から，本研究の提案手法による類似度は作曲未経験者よりも作曲経験者が感じる類似度に近い類似度が算出されていると考えられる．

6. おわりに

本研究では「ピッチクラスの割合」，「調性に基づいた和音の機能」，「和音の遷移」が考慮された類似度を楽曲類似度と定め，和音の分散表現の獲得及び，楽曲類似度算出手法の提案を行った．楽曲ファイルを和音列へ変換し，調性を推定することで和音の機能を一意に定めた度数表記へ変換し，Chord2Vec の学習データとして用いることで，和音の分散表現を獲得した．また，類似度を算出したい楽曲ファイルを度数表記へ変換し，N-gram カウントを行った後，得られた分散表現を用いた比較を行うことで，楽曲間類似度を算出した．結果，提案手法により算出された類似度は人間が聴取した際に感じる類似度と近い類似度であるという結果が得られた．したがって，本研究の提案手法による楽曲類似度は，類似楽曲の推薦に貢献できると考える．

今後の展望として，対象曲数を増やし，本手法の一般性を確認する必要がある．さらに，和音に加え「旋律(メロディ)」「律動(ビート)」を考慮した類似度の算出手法を検討することが挙げられる．特に，「律動」に焦点を当てることで，時系列に基づいた楽曲の構造を考慮した類似度の算出ができることが期待される．

参考文献

[1] 島岡 譲. “和声—理論と実習 (1)”. 音楽之友社, 1964
[2] Fujishima, T. “Realtime chord recognition of musical sound: a system using Common Lisp Music” ICMC, Beijing, China, pp. 464–467, 1999.
[3] Y. Wu and W. Li, “Automatic Audio Chord Recognition with MIDI-Trained Deep Feature and BLSTM-CRF Sequence Decoding Model.” in IEEE/ACM Transactions on Audio, Speech, and Language Processing, pp.355-366, 2018.
[4] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, “Efficient Estimation of Word Representations in Vector Space.” CoRR Vol. abs/1301.3781, 2013.
[5] Xin Rong, “word2vec Parameter Learning Explained” CoRR Vol. abs/1411.2738, 2014.
[6] Z. Harris. “Distributional structure.” Word, 10(23), pp.146–162, 1954.
[7] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. “Distributed representations of words and phrases and their compositionality.” In NIPS 26, pp.3111–3119, 2013.
[8] Free Music Archive http://freemusicarchive.org/, 参照 2018 年 5 月 30 日