Vol.2020-MUS-127 No.27
Vol.2020-SLP-132 No.27
2020/6/7

Doc2vecの応用によるメロディ検索

平井 辰典[1,a)]

概要：メロディは音楽における最も象徴的な要素の一つであり，楽曲を特徴づけるための大きな役割を担うものである．そのため，楽曲検索システムにおけるメロディに基づいた検索機能の実現は重要な課題であるといえる．本研究では，自然言語処理分野においてLeとMikolovによって提案された文章の分散表現であるParagraph vector（通称doc2vec）を音楽のメロディに応用することで，メロディ検索を実現する．これまでにHiraiとSawadaにより提案されたmelody2vecでは，メロディのセグメンテーションを行うことによってメロディ内のフレーズの分散表現を獲得し，類似フレーズの検索やメロディ内のフレーズの置換による再編集などを実現した．本稿では，メロディ内のフレーズに対する処理に限定されていたmelody2vecを1曲分のメロディに対する処理へと拡張するためにdoc2vecによるモデル化を行う．またその応用として，メロディ検索を含むいくつかの活用事例を提案する．

1. はじめに

音楽においてメロディは楽曲を特徴づける役割を担う重要な要素である．本稿では，自然言語処理分野で提案されている文章の分散表現を実現する手法であるdoc2vecの応用によるメロディ検索手法を提案する．

現在多くの音楽サービスにおいて主流の楽曲検索は，曲名やジャンル，歌詞などのメタデータを用いた検索手法である．そのような中で音楽情報検索（MIR）分野では，音響信号処理や楽譜の記号処理を用いた楽曲検索の試みが活発に研究されている．音楽情報検索研究分野のトップ国際会議であるISMIRでは，MIREX（Music Information Retrieval Evaluation eXchange）と呼ばれる音楽情報検索のコンペティションが開催されており，過去にはメロディを用いた楽曲検索のタスクとしてSymbolic Melody Similarityに関するコンペティションも行われていた（2005年-2015年）．

メロディによる楽曲の検索が実現することで，曲のタイトルなどのメタ情報を知らずとも楽曲の検索ができることが期待される．本来，音楽は音を楽しむメディアであり，メタデータは音楽本体に付随する付加情報にすぎない．そのため，音楽というメディアにおいてコンテンツの実体を用いた本質的な検索を実現するには，音楽そのものを構成する要素による楽曲検索手法が必要であると考えられる．一方で音には膨大な情報が含まれており言語化することが困難なため，便宜上メタデータが最も扱いやすい情報となっており，メタデータを用いた検索が主流となっているのが現状である．音楽を構成する要素の中でも，メロディは楽曲を特徴づける重要な役割を担い，かつ，比較的記憶に残りやすい音楽要素であると考えられる．曲名はわからずとも鼻歌では歌うことができる曲は，個人差はあれど少なからずあるのではないだろうか．既存の検索システムとして，鼻歌をクエリとして目当ての楽曲を探し当てるシステムは提案されているが，その使用用途として，目当てとなる検索対象の楽曲が明確に存在する場合の使用が想定されている．そこで，より柔軟に似ているメロディを探索したい場合などを想定して，メロディに基づく検索の実現について検討する．

本稿では，メロディ検索手法の新しいアプローチとして，ニューラルネットワークベースのメロディの分散表現を用いた手法を提案する．それにより，類似メロディの検索や，メロディのベクトル空間内での演算による平均メロディの検索などの新たなメロディ検索のアプリケーションの実現についても検討する．

2. 関連研究

2015年まで実施されていたMIREXのSymbolic Melody Similarityタスク（SMS）は，5000曲以上のフォークソングに関するデータセットの中からクエリに対して類似度が高いメロディを検索し，その結果を人が実際に聴いて似ているかどうかを評価するというものであった．参加チームの減少から，現在のMIREXではSMSのタスクは実施されていないが，2015年までに実施されていた参加チームの提案手法の多くは，独自の評価尺度を提案するアプローチ

[1] 駒澤大学
Komazawa University
[a)] thirai@komazawa-u.ac.jp

や，独自のデータ表現によってメロディを扱った上で類似度として数値化する手法などに分類できる．例えば，直近に開催された 2015 年の MIREX における SMS で最も優れた結果を出した Urbano による幾何学的なアプローチが挙げられる [1]．このアプローチは，メロディを時間とピッチの軸で表現された平面における幾何学的な曲線であるとみなし，メロディ間のアライメントを通じてその類似度を算出する手法である．このように，SMS ではメロディの表現手法や評価尺度の検討が主なアプローチであった．一方で，近年情報科学分野において著しい成果を挙げている（深層）ニューラルネットワークベースのアプローチについてはこのコンペティションが開催されていた時点では提案されていなかった．

記号として表現されたメロディをニューラルネットワークベースのアプローチによってモデル化した手法として，Hirai と Sawada の melody2vec [2] が挙げられる．Melody2vec は，2013 年に Mikolov らが提案して自然言語処理分野で大きな成果を挙げた単語の分散表現手法 word2vec [3] をメロディへと応用したものである．この手法では，GPR というGTTM [4] におけるメロディのグルーピング規則を基にメロディをフレーズに分割したものを自然言語における単語と見立て，word2vec において提案された skip-gram という手法でメロディ内のフレーズをモデル化している．これにより，メロディに含まれるフレーズをニューラルネットワークベースのアプローチによってモデル化することに成功したが，melody2vec ではメロディ全体のモデル化はできていなかった．

同様に，word2vec は単語のベクトル化は実現しているが，文章全体のベクトル化については，その後 Le と Mikolov によって提案された文章の分散表現である paragraph vector によって実現された [5]．Paragraph vector は，word2vec に対応して doc2vec という通称でも呼ばれており，一文の文章のベクトル化から複数の文章の集まりであるニュース記事などのベクトル化までも実現している．そこで本稿では，word2vec のメロディへの応用による melody2vec の実現と同様に，doc2vec のメロディへの応用について検討する．

3. データの準備

ニューラルネットワークベースのモデル化のアプローチにおいて，データの準備は重要なプロセスである．用意するデータの量は多ければ多いほど好ましいため，本稿では melody2vec で使用された約 10,000 曲分のメロディデータを利用する．Melody2vec では，インターネット上に存在する midi ファイルをクローリングすることによって収集された 176,581 曲分の楽曲データセットである the Lakh dataset を利用し，メロディトラックが抽出できた 10,853 曲分のデータを学習に用いていた．本稿でも，同じデータを用いたが，その中に含まれるメロディのデータが空ではない 10,738 曲分のデータを学習に用いた．

The Lakh dataset の MIDI データは，インターネット上に存在する不規則な midi ファイルから構成されているもので，トラックの数や，その正確さについては担保されていないものである．そのため，プログラムによって陽にメロディのトラックが存在すると判定できるファイルは約 17 万曲のうちの約 1 万曲程度である．本稿では，なるべく確実にメロディのデータを扱うために，約 16 万曲分のメロディに関する情報が不確実な MIDI ファイルを利用しないこととしている．また，上記の約 1 万曲分のメロディデータに関しても，空のメロディトラックが存在していたり，メロディトラックとはいいながらも別の楽器のトラックであったりと，楽譜通りに正確にメロディを記録しているようなデータばかりではない．The Lakh dataset には同じ楽曲の MIDI ファイルも含まれているが，アレンジの違いや MIDI ファイル作成者の作成方法の違いなどにより，完璧に同じ演奏情報を含んだ MIDI ファイルは著者が観測した範囲では存在しない．

用意した 10,738 曲分のメロディデータに対して doc2vec を適用するに際して，メロディを文章とみなした際の「単語」にあたる概念を定義しなければならない．本稿では，メロディにおける単語にあたる概念として melody2vec で提案されたメロディのフレーズを採用する．Melody2vec におけるメロディのフレーズは，GPR に基づいてメロディを分割することによって得られたものである．

メロディのセグメンテーションの処理に加え，調の正規化処理を行い，すべてのメロディの調を合わせる．様々な調のメロディが含まれている状態では，例えば同じ “C-D-E” というフレーズでも，ハ長調のメロディにおける “C-D-E” とヘ長調における “C-D-E” では役割が違うためである．調の正規化は，melody2vec で提案されたヒストグラムに基づく調推定手法によって推定した調を基に転調をすることで実施し，長調の場合はハ長調（C）に，単調の場合はイ単調（Am）となるように転調する．また，調の正規化と同時にオクターブの正規化も行い，メロディの頻出オクターブが C4 のオクターブとなるように調整する．

上記の処理は，基本的には melody2vec と同様の前処理となっているため，その詳細については melody2vec の文献 [2] を参照されたい．Doc2vec の応用にあたって異なる点は，Paragraph vector の学習のために必要なドキュメント ID に相当する ID をメロディ 1 曲毎に付与しているという点だけである．

4. メロディベクトルのモデル化

Doc2vec は，PV-DM または PV-DBOW というニューラルネットワークモデルによって文章の分散表現である paragraph vector を学習する手法である．PV-DM（Para-

表 1　学習パラメータ

	次元	窓幅	negative sampling	min count	エポック
PV-DM	10/300	5	5	2	300
PV-DBOW	10/300	15	5	2	300

graph Vector: A distributed memory model）は，注目する単語の周辺の単語とドキュメント ID を用いて注目単語を予測するような中間層の重みを学習するモデルである．このときの中間層の重みこそが求めたいベクトルとなる．PV-DBOW（Paragraph Vector without word ordering: Distributed bag of words）は，ドキュメント ID を入力として，文章に含まれる単語を予測するような中間層の重みを学習するモデルである．PV-DBOW では単語の順序については考慮しない．

本稿では，メロディ用の doc2vec の学習に，python の自然言語処理ライブラリである gensim を用いた．Doc2vec の学習に際して，窓サイズや negative sampling などのパラメータの指定が必要となるが，著者が複数のパラメータの組み合わせの学習を通して実験的に選択した数値を採用した．採用したパラメータの詳細を**表 1** に示す．表中の min count は，その値以下の出現回数の単語については学習時に考慮しないことを表すパラメータであり，表 1 の値の場合，データ全体を通じて 1 度しか出現していない単語については考慮しないことを表す．ベクトルの次元については，それぞれの手法で 10 次元及び 300 次元のモデル化を行った．

5. 結果

ここまでに記述した方法によって，楽曲のメロディを 10 次元もしくは 300 次元のベクトルへと変換するようなメロディの embedding（埋め込み）モデルが実現した．これにより，データセット内の任意のメロディをクエリとして，そのメロディに近い別のメロディを検索することが可能となった．一方でその妥当性については検証をしなければならない．

メロディに関するベクトル表現が実現したことで，10,738 曲分のメロディについて，データセット内の他のメロディとの類似度，計 115,293,906 ペア分（$10,738 \times 10,737$）を算出することができる．しかし，これらすべてのメロディ対の類似度が実際に人間が感じるメロディ同士の近さに対応するかを評価することは困難である．そこで本稿では，the Lakh dataset には，同一楽曲のデータが複数含まれていることがあることを利用した評価を行う．メロディベクトルの有効性を簡易的に判定するために，115,293,906 ペアの類似度の上位 10 ペアの中に同じ曲のメロディが何ペア含まれているかを検証した．また，同時に下位 10 ペアの中に同じ曲のメロディが含まれていないかについても実際に聴取することで確認した．ここで，ベクトル同士の類似度を測る尺度としては，コサイン類似度を採用した．

表 2　同一楽曲のメロディ含有率

手法	上位/下位	10 ペアに占める同一曲ペア数
PV - DM 10	上位	3
	下位	0
PV - DM 300	上位	3
	下位	0
PV - DBOW 10	上位	5
	下位	0
PV - DBOW 300	上位	6
	下位	0

検証の結果を**表 2** に示す．PV-DM 10 は 10 次元のベクトル，PV-DM 300 は 300 次元のベクトルのモデル化を行った場合であり，PV-DBOW についても同様である．表 2 の結果の通り，類似度が高い上位のペアの中には同一楽曲のペアが含まれているが，下位のペアには含まれていなかった．115,293,906 ペアのうち，同一楽曲のペアがどれだけ含まれているかについては把握する術がないが，10 ペアは優に超えている．通常，midi ファイルの曲名がわかっていれば同じ曲であるかの判定は非常に簡単である．しかし，the Lakh dataset の楽曲の多くは，曲名についてのメタデータが付与されていないことが多く，付与されていたとしてもファイル名の一部として不規則な形式で記述されていることが多い．また，仮に曲名を用いた検索をする場合，同じ曲であるか否かの検索しかできないため，本研究で実現したいこととは趣旨がずれるものである．

本評価において，まったく同じ曲の同じメロディであれば距離は 0 となり，類似度が高くなることは当然のように考えられるものであるが，the Lakh dataset の MIDI ファイルは，インターネット上に存在していた wild なデータであり，全ペアの中に距離が 0 となるペア（完全に同じメロディのペア）は存在しなかった．たとえ同じ楽曲であってもメロディの細部や発音タイミングなどが異なっており，必ずしも類似度が高いと判定できるような綺麗なデータではないということである．

一方で，the Lakh dataset 内には，表 2 に示された数よりも多くの同一楽曲の midi ファイルが含まれており，それらのペアが必ずしも上位に集中しているわけではないという点については，本手法がまだ十分に同一楽曲を似ていると判定しきれていないことを示している．この点については，今後さらなる精度の向上を図れるようなモデル化の実現を目指していきたい．

表 2 において，同一楽曲のメロディであるかどうかは，著者がペアのメロディを聴き比べることでラベリングした．著者の主観ではあるが，下位のペアについてはメロディが異なる楽曲のものであることが比較的わかりやすかったため，ラベリングが簡単であった．一方で上位のペアについ

表 3 類似メロディ検索（メロディベクトル：10 次元）

クエリ	D♯4:1/16 - D♯4:1/16 - D♯4:1/8 - D♯4:1/16 - F4:1/6 - C4:1/8 - D♯4:1/8 - D♯4:1/8 - G♯4:1/16 - A♯4:1/6 - C5:1/16 -		
手法	順位	類似度	メロディ
PV - DM	1	0.951	D♯5:1/16 - R:1/4 -G4:1/16 - R:3/8 - G♯4:1/8 - D♯5:1/16 - R:1/4 - D5:1/16 - R:1/4 - ....
	2	0.854	C5:1/16 - D♯5:1/8 - D♯5:1/8 - D♯5:1/8 - D♯5:1/16 - D♯5:1/6 - D♯5:1/16 - C5:1/16 - ....
	3	0.846	D♯4:1/2 - D4:1/2 - C4:3/4 - R:1/6 - D♯4:1/2 - F4:1/4 - G4:1/8 - F4:1/4 - D♯4:3/4 - ....
	10,735	-0.842	A♯3:3/8 - G♯3:1/8 - G♯3:3/4 - R:3/4 - A♯3:3/8 - G♯3:1/8 - G♯3:3/4 - R:3/4 - F4:3/8 - ....
	10,736	-0.901	G♯3:1/8 - G♯3:1/8 - G♯3:1/8 - C4:1/8 - D♯4:1/6 - D♯4:1/4 - D♯4:1/16 - D♯4:1/16 - D♯4:1/8 - ....
	10,737	-0.928	A♯3:1/8 - R:1/4 - A♯3:1/2 - R:3/8 - A♯3:1/4 - C4:1/8 - D4:1/6 - R:1/6 - D♯4:1/4 - R:1/6 - ....
PV - DBOW	1	0.942	D♯4:4/4 - F4:4/4 - D♯4:3/4 - R:4/4 - D♯4:4/4 - F4:4/4 - D♯4:4/4 - R:3/8 - C4:1/12 - ....
	2	0.941	F4:3/4 - F4:1/12 - A♯3:1/16 - F♯4:3/4 - F♯4:1/12 - F4:3/4 - F4:1/12 - C4:1/2 - C4:1/16 ....
	3	0.905	F4:1/8 - F4:1/8 - D♯4:1/12 - D♯4:1/6 - F4:1/4 - F♯4:1/4 - F4:1/4 - D♯4:1/8 - F4:1/4 - ....
	10,735	-0.822	C♯4:1/8 - F4:1/4 - F4:1/8 - F4:1/8 - F4:1/4 - D♯4:1/8 - C♯4:1/8 - D♯4:1/4 - D♯4:1/8 - ....
	10,736	-0.823	F4:1/16 - D4:1/16 - F4:1/6 - R:4/4 - D4:1/16 - C4:1/16 - D4:1/4 - G3:1/16 - F3:1/4 - ....
	10,737	-0.829	D♯4:1/4 - G4:1/4 - G♯4:4/4 - C5:1/6 - G4:3/4 - R:1/6 - F4:1/6 - G4:1/12 - G♯4:1/12 - ....

表 4 類似メロディ検索（メロディベクトル：300 次元）

クエリ	D♯4:1/16 - D♯4:1/16 - D♯4:1/8 - D♯4:1/16 - F4:1/6 - C4:1/8 - D♯4:1/8 - D♯4:1/8 - G♯4:1/16 - A♯4:1/6 - C5:1/16 -		
手法	順位	類似度	メロディ
PV - DM	1	0.821	C♯5:1/8 - C5:1/4 - C5:1/16 - C5:1/6 - A♯4:1/8 - A♯4:1/8 - G4:1/8 - A♯4:1/4 - G4:1/8 - ....
	2	0.817	C♯5:1/8 - C5:1/4 - C5:1/16 - C5:1/6 - A♯4:1/8 - A♯4:1/8 - G4:1/8 - A♯4:1/4 - G4:1/8 - ....
	3	0.797	F4:1/16 - G♯4:1/16 - F4:1/16 - G♯4:1/4 - F4:1/16 - G♯4:1/16 - F4:1/16 - C5:3/8 - R:3/4 - ....
	10,735	-0.248	F4:1/12 - F4:1/6 - G4:1/12 - G♯4:1/6 - G4:1/12 - G♯4:1/4 - R:1/6 - G4:1/12 - G♯4:1/4 - ...
	10,736	-0.275	A♯3:1/6 - G3:1/12 - A♯3:1/6 - G3:1/12 - A♯3:1/6 - A3:1/12 - G3:1/4 - F3:1/6 - G3:1/12 - ....
	10.737	-0.326	G4:1/6 - A♯4:1/12 - G4:1/16 - G4:1/8 - A♯3:1/4 - R:1/4 - A♯3:1/12 - C4:1/6 - D♯4:1/12 - ....
PV - DBOW	1	0.414	C♯5:1/8 - C5:1/4 - C5:1/16 - C5:1/6 - A♯4:1/8 - A♯4:1/8 - G4:1/8 - A♯4:1/4 - G4:1/8 - ....
	2	0.404	C♯5:1/8 - C5:1/4 - C5:1/16 - C5:1/6 - A♯4:1/8 - A♯4:1/8 - G4:1/8 - A♯4:1/4 - G4:1/8 - ....
	3	0.370	F4:1/16 - G♯4:1/16 - F4:1/16 - G♯4:1/4 - F4:1/16 - G♯4:1/16 - F4:1/16 - C5:3/8 - R:3/4 - ....
	10,735	-0.166	F♯3:1/16 - F♯3:1/8 - G♯3:1/16 - G♯3:1/8 - B3:1/16 - B3:1/8 - B3:1/16 - B3:1/16 - B3:1/16 - ....
	10,736	-0.167	G5:1/16 - G5:1/16 - G5:1/16 - G5:1/16 - G5:1/16 - E5:1/8 - A5:1/12 - A5:1/16 - G5:1/2 - ....
	10,737	-0.192	B4:1/12 - R:3/8 - B4:1/12 - B4:1/12 - R:4/4 - B4:1/16 - B4:1/8 - B4:1/8 - B4:1/8 - B4:1/16 - ....

ては，メロディ同士に類似する箇所が多くあると感じ，同一の楽曲であるかどうかをラベリングするための難易度が高く感じた．このことを評価するために，今後，主観評価実験を実施することを検討している．

6. 応用例

本章では，提案手法によって獲得したメロディベクトルの応用例をいくつか紹介する．

6.1 類似メロディ検索

メロディのベクトル表現が実現したことにより，ベクトル同士の距離計算によって類似メロディを検索することが可能となる．クエリとなる任意のメロディのベクトルに対して，コサイン類似度が高いメロディをデータセットから検索することで類似メロディが求められる．類似度が高い上位のメロディを提示することで，似ているけれど違うメロディを探すことが可能となる．**表 3** 及び**表 4** は，クエリとなったメロディの冒頭部と類似度が高かった上位 3 つ及び低かった下位 3 つのメロディの冒頭部及びそれらの間のコサイン類似度を示したものである．クエリに対して，10 次元及び 300 次元のメロディベクトルを用いて，PV-DM，PV-DBOW でモデル化した場合の上位及び下位のメロディを示している．この表において，1/4 は 4 分音符，1/8 は 8 分音符，1/16 は 16 分音符，R は休符を表している．検索対象はすべて転調処理及びオクターブの正規化処理を行った後のメロディとしている．

表 4 は 300 次元のメロディベクトルを用いた場合の結果であるが，PV-DM 及び PV-DBOW の結果で，類似度の値こそ異なっているが，1 位から 3 位までの曲が同一であった．特に，1 位と 2 位の曲は同じ曲であった．このことから，ベクトルの次元数を 300 次元と大きくしたことで，モデル化の手法が違いながらも，似たようなデータの偏りを持つベクトルが得られたのではないかと推察される．

クエリは The Beatles の Let it be のメロディであり，元々がハ長調の楽曲であるが，調の正規化処理を行った結果，ハ長調からずれてしまっている．紙面の都合上表 3，表 4 にはメロディの冒頭の一部しか掲載できておらず，実際にこれらのメロディが似ているか否かを判断するために十分なデータが表示されているわけではないことに注意されたい．本手法では，あくまでもメロディ全体をモデル化

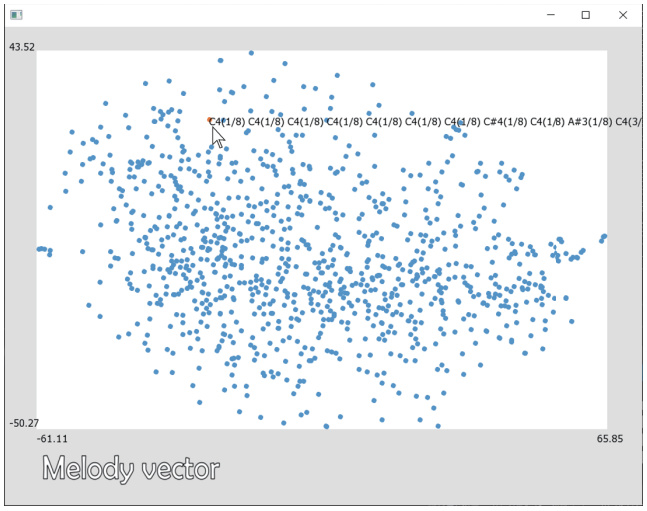

図 1　メロディ平面上のメロディ探索インタフェース

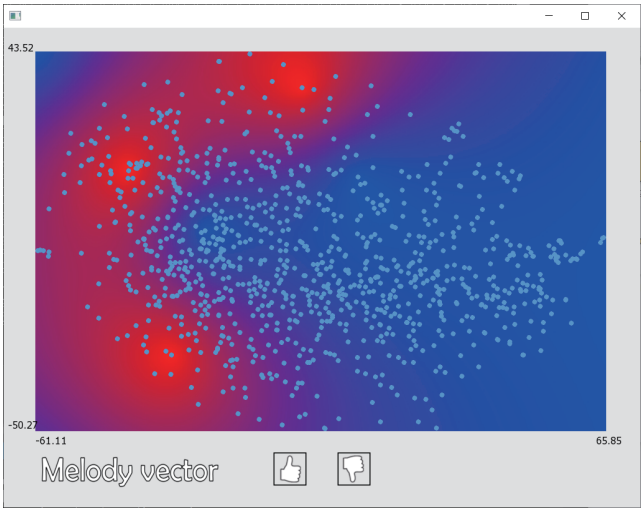

図 2　メロディ平面上の好みの領域の可視化

している．5 章でも述べたように，本メロディベクトルを用いた類似度の妥当性については，今後の主観評価実験によって検証する予定である．また，データセット内には，同じ Let it be の midi ファイルが複数含まれていたが，それらとの類似度はあまり高くはなく，類似度の上位には含まれていないという結果となった．この点については，今後より踏み込んだ分析が必要となる．

6.2　メロディ平面上のメロディ探索

メロディのベクトルにより，メロディ空間が構築できるが，インタフェース上で高次元空間を扱うことは難しいため，二次元平面へのマッピングを行う．マッピングすることで，高次元のメロディ空間を人が認識しやすいメロディ平面に落とし込むことができる．本稿では，二次元のマッピング手法として t-SNE [6] を採用する．

メロディ探索を実現するために，データセットの一部（1,000 曲分）のメロディデータを t-SNE によって二次元平面にマッピングしたものを，クリックしながら試聴できるインタフェースを実装した．インタフェースの画面の様子を**図 1** に示す．1 つ 1 つのプロットが 1 曲のメロディを表しており，距離が近いプロット同士は，メロディベクトル間の距離が近い．ここでは，PV-DBOW による 300 次元のモデル化の結果から，t-SNE による二次元マッピングを行い，10,738 曲分のプロットの中からランダムに選んだ 1.000 曲分のメロディのプロットのみを表示している．インタフェース上でプロットにマウスオーバーすると，選択されたメロディがテキスト形式で表示される．本来，楽曲名が表示できると望ましいが，the Lakh dataset の MIDI ファイルに関するメタデータは楽曲名が特定できる形で整理されていないためこのような形式をとっている．

6.3　好みのメロディ領域のモデル化

メロディベクトルの実現により，ユーザの好みのメロディの領域という概念を見つけられる可能性がある．まず，6.2 節で紹介したインタフェースを用いて探索したメロディがユーザの好みのメロディであるか否かをラベリングできるようにする．ラベリングされた複数のメロディの座標と，好みに関するラベル（0，1）を用いることで，好みのメロディが存在する領域を可視化する方法について検討する．ここでは，RBF 補間によりユーザの好みに関するラベル情報の補間を行い，メロディ平面上にヒートマップを作成する例を挙げる．

図 2 にユーザの好みのメロディ領域を可視化した例を示す．図中のメロディ平面内の赤い領域は，ユーザが好みであるとラベリングしたメロディに近い領域である．RBF 補間は，任意の点からの距離のみに依存した値の補間が可能であり，ここでは「好み（1）」のラベルが付与されたプロットからの距離に応じて値が減衰するような補間を行っている．RBF 補間における放射基底関数には，式 (1) に示す多重二乗 RBF を採用した．

$$\phi(\boldsymbol{x}) = \sqrt{1 + (\epsilon||x||)^2} \tag{1}$$

メロディ平面上で，ユーザがメロディを試聴しながら，気に入ったメロディがあればインタフェース下部の like ボタンを押すことによりヒートマップの計算を行い，その可視化結果をメロディ平面に反映させる．また，dislike ボタンも使用することで，メロディ平面上の好みの領域を絞っていくこともできる．

本節では，t-SNE を用いて二次元に圧縮したメロディ平面における領域のヒートマップを示したが，この状態では元の 300 次元のベクトルの向きなどの情報は保持できていない．そのため，プロット同士の近さが元のベクトルの近さには相当するものの，平面の右側や左側といった領域に関してはあまり意味がない状態となっている．この問題については，今後より有効な可視化手法の導入により改善していきたいと考えている．現状では本機能の有効性につい

Vol.2020-MUS-127 No.27
Vol.2020-SLP-132 No.27
2020/6/7

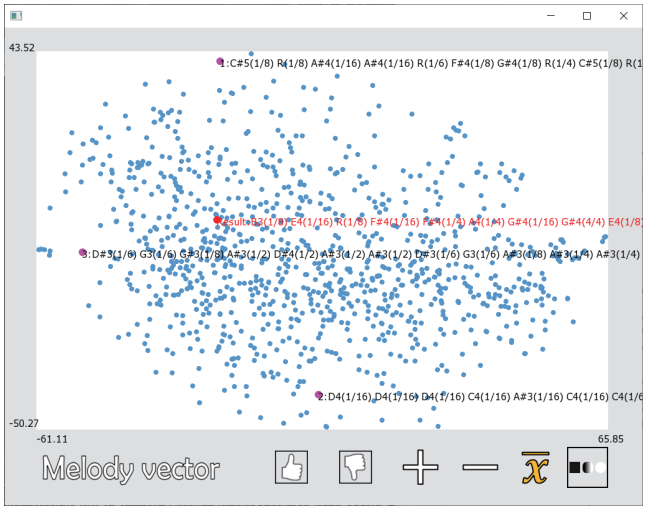

図 3 平均メロディの探索

ての検証はできていないが，本稿で提案したメロディのベクトル化によって，このような直感的なアプローチによるメロディ検索が実現できる可能性があると考えている．

6.4 メロディベクトルの演算による平均メロディの探索

メロディがベクトルとして表現できるということは，ベクトルの演算を行うことができるということである．Word2vec で提案された単語ベクトルの演算（例：“king” - “man” + “woman” ≈ “queen”）をメロディでも実現可能である．複数のメロディベクトルの加算や減算，平均ベクトルの算出，ベクトル A からベクトル B へのモーフィングなどが実現できる．ここでのモーフィングとは，ベクトル A とベクトル B を結ぶ線分の近傍に存在するメロディを順に再生していくことで，メロディ平面において徐々にメロディを切り替えていくという実装である．

図 3 には，3 曲分のメロディベクトルの平均ベクトルを算出し，その最近傍に位置するメロディを探索している様子を示す．メロディ平面上の上部，下部，左部分に位置する紫色でプロットされたメロディの平均ベクトルに最も近かったメロディのプロットが中央付近の赤色のメロディである．ここで，ベクトル演算は t-SNE の二次元座標を使っての演算ではなく，元の 300 次元のメロディベクトルを対象にした演算を行っている．そのため，平面における見かけ上の平均の位置と実際の演算結果との位置が対応しているわけではない．

図 3 のインタフェースの右下に示されている演算子に対応するボタンを押すことでベクトルの演算が可能である．この機能の実装自体は，単純なベクトルの演算と最近傍探索の実装のみで実現できるため非常に簡単であるが，その効果の検証については困難であり，本研究の今後の課題として挙げられている．特に，メロディには誰もが共有可能な明確な意味というものが内在されているわけではない．そのため，抽象的な概念同士の演算となってしまい，演算の結果得られたメロディが果たして正しいのかを検証することは難しいと考えている．今後，様々な側面からの評価を通して，メロディそのものの表現，評価についての探究を行っていきたい．

以上のように，メロディベクトルの実現によってこれまでに挙げた様々なアプリケーションが実現できる．一方で，その機能が妥当かどうかについてはさらなる検証が必要である．

7. 議論

本稿では，doc2vec の応用によるメロディの分散表現手法を提案し，ニューラルネットワークベースのモデル化によるメロディの検索を実現した．本提案手法により，メロディのベクトル空間の構築が実現した．メロディのベクトル空間については，今後さらなる追究を行っていく予定である．例えば，6 章で応用例として示したように，個人が好きなメロディがベクトル空間内でどのように偏っているかや，より好みのメロディの探索方法についてなど，さらなる探究の余地があると考えている．また，提案したメロディベクトルの妥当性についてはより踏み込んだ検証が必要になると考えており，今後の研究を通してさらなる評価を行っていく予定である．

本研究は，word2vec のメロディへの応用である melody2vec を基にモデル化を行っている．そのため，word2vec の欠点として挙げられるボキャブラリーに含まれない未知語への対応ができないという点についても同様に引き継いでしまっている．Word2vec のそのような欠点に対応するための手法として，自然言語処理分野では fastText [7] などの手法が提案されている．FastText は単語の構成要素である subword を導入することで，ある程度の範囲で未知語に対応することができる．Melody2vec についても fastText と同様のアプローチで submelody を導入して対応することが可能であると考えられる．

また，2018 年に提案された BERT [8] を始め，自然言語処理分野は深層ニューラルネットワークを用いた手法により着実な進歩を遂げている．自然言語処理における成功例は音楽情報処理においても有効であることが多いと考えられ，今後その可能性をより深く追究していきたい．

謝辞 本研究は JSPS 科研費 JP 19K20301 の助成を受けたものである．

参考文献

[1] Urbano, J.: MelodyShape at MIREX 2015 Symbolic Melodic Similarity, *Music Information Retrieval Evaluation eXchange* (2015).
[2] Hirai, T. and Sawada, S.: Interactive Music Summarization based on Generative Theory of Tonal Music, *Journal of Information Processing*, Vol. 27, pp. 278–2867 (2019).
[3] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S.

and Dean, J.: Distributed Representations of Words and Phrases and Their Compositionality, *Advances in neural information processing systems*, pp. 3111–3119 (2013).

[4] Lerdahl, F. and Jackendoff, R. S.: *A Generative Theory of Tonal Music*, The MIT press (1983).

[5] Le, Q. and Mikolov, T.: Distributed Representations of Sentences and Documents, *Proceedings of the International Conference on Machine Learning*, pp. 1188–1196 (2014).

[6] Maaten, L. v. d. and Hinton, G.: Visualizing Data using t-SNE, *Journal of machine learning research*, Vol. 9, No. Nov, pp. 2579–2605 (2008).

[7] Joulin, A., Grave, É., Bojanowski, P. and Mikolov, T.: Bag of Tricks for Efficient Text Classification, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 427–431 (2017).

[8] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 4171–4186 (2019).