

# GPRに基づくメロディセグメンテーションによる フレーズの分散表現の獲得

平井 辰典<sup>1,a)</sup> 澤田 隼<sup>2</sup>

**概要:** 本稿では、自然言語処理分野において大きな成功を取めている word2vec のフレームワークをメロディへと拡張することを目的とした処理手法を提案する。具体的には、GTTMにおける音符のグルーピングに関する規則である GPR を用いてメロディをセグメンテーションすることで、メロディにおける単語に相当するメロディフレーズを定義し、フレーズの分散表現を獲得する。10,853 曲分の MIDI データから抽出したメロディデータに対して本手法を用いてフレーズの分散表現を獲得するためのモデルを構築した。また、主観評価実験によって本手法により構築したモデルを評価することで、その有効性を示した。さらに、本手法をメロディ編集に応用した例を通して、音楽情報処理におけるメロディフレーズの分散表現の可能性を示す。

## 1. はじめに

メロディは音楽において最も重要な要素の一つである。計算機を用いてメロディを扱う手法はいくつか提案されており、大きく二つに分けることができる。音符等の楽譜情報を用いて直接メロディを扱う記号処理によるアプローチと、音声信号に起因する特徴量等を用いて間接的にメロディを扱う信号処理によるアプローチである。記号処理によるアプローチでは、信号処理では直接扱うことが難しいメロディ情報そのものを扱うことができる。例えば、GTTM[1] や IRM[2], [3] が代表的な記号処理により、メロディの構造や意味にまで踏み込んだ分析をする手法である。中でも GTTM 分析の結果得られる木構造は、自然言語の構文木に対応した形でメロディを扱う手法である。

記号処理が最も活発に研究されている分野として、自然言語処理が挙げられる。自然言語処理の研究で提案されている手法を音楽情報処理に応用している事例は数多く挙げられる。しかし、自然言語処理分野の中でも近年最も成功している手法である word2vec[4] をメロディに応用した事例はまだ報告されていない。Word2vec は Mikolov らによって提案された、単語を数値ベクトルにより分散表現するための手法であり、記号表現である単語を数値表現であるベクトルへと変換することを可能としている。

メロディは複雑な概念であり、従来の手法 [5] では直感的、潜在的な類似度を計算することは困難であった。Word2vec が応用できれば、メロディのベクトル表現によりメロディの意味的な関係を記述することが可能となるかもしれない。さらには、メロディの定量的な評価の実現にもつながる可能性もある。メロディは音楽において基本的な要素であるため、メロディの分散表現を獲得することは、音楽情報検索分野の今後の発展につながる価値のある課題であると考えられる。

Word2vec のアプローチを応用してメロディのベクトル表現を実現する上で、最も大きな課題はメロディと単語の構造の違いである。その違いは、word2vec のフレームワークをそのままメロディに応用することができないことの原因となっている。Word2vec をメロディに応用するためには、まずメロディにおける「単語」を定義しなければならない。本研究では、GTTM における自然言語とメロディとの対応関係を用いることでメロディにおける「単語」に対応する要素を取得する。具体的には、メロディ内のフレーズを単語に対応する概念と見立て、GTTM における音符のグルーピングに関する規則である GPR (Grouping Preference Rules) を利用してメロディをフレーズへとセグメンテーションする。

本稿では、word2vec のフレームワークのメロディへの拡張である melody2vec を提案し、構築したモデルの評価やアプリケーションへの応用を通じてその可能性について議論する。

<sup>1</sup> 駒澤大学  
Komazawa University

<sup>2</sup> 公立はこだて未来大学  
Future University Hakodate

a) thirai@komazawa-u.ac.jp

## 2. 関連研究

自然言語処理と同様のフレームワークで音楽を扱う手法はこれまでに多く提案されてきている。特に、GTTMは音楽の分析に関する規則を記述した理論であり、GTTMに関連した多くの研究がなされている [6], [7]。GTTMでは、タイムスパン木によりメロディを記号的に分析することを可能としている。しかし、音楽のように多くの例外が発生するドメインにおいてルールベースの処理手法は必ずしもうまくいくとは限らない。例えば、GTTMの中の複数のルールが競合する事例などもあり、その優先順位はコンテキストに依存することがある。そのため、データに対して柔軟な処理を実現する手法が望まれる。

Word2vec[4]は学習データから得られる単語と文脈との関係を学習する手法として、自然言語処理分野において著しい成果を挙げている。Word2vecでは、大規模な文章データの入力を基にして、類似した意味や似たような使われ方をする単語のembedding(埋め込み)をskip-gramやCBoW(Continuous Bag-of-Words)と呼ばれる手法によって学習する。これらの手法により単語の意味的な関係の効果的な学習が実現する。もしもこれらの手法に似たようなフレームワークでメロディを扱うことができれば、word2vecの応用によってもたらされた自然言語処理分野の数々の手法が音楽情報処理分野においても応用できると考えられる。

Word2vecと類似したアプローチを様々なドメインに派生させた研究は多く提案されている [8], [9], [10]。Illustration2vec[8]は、CNNによって二次元イラスト画像のベクトル表現を取得している。この手法では、イラストのアノテーションの推定や意味的に類似する画像の検索、二つのイラストの意味的なモーフィングを実現している。Word2vecのアプローチは、音楽のドメインにも応用されている。Music2vec[10]は、楽曲を聴く順番に関するコンテキスト情報を基に楽曲のベクトルを学習する。学習されたベクトルは対象となるユーザの音楽の好みを反映しており、楽曲推薦などのタスクにおいて有効である。

Herremansらは、ポリフォニー音楽における楽譜を等間隔にスライスし、各瞬間の和声構造を単語に見立てて、word2vecの枠組みにより分散表現を獲得している [11]。この手法は、文脈情報を学習するという点でメロディも考慮したモデル化を行っているが、各瞬間の旋律間の繋がりに関する分散表現を獲得しているもので、メロディについてのベクトル表現を実現しているわけではない。また、メロディを構成する各音符とそのメタ情報を単語と仮定してLSTMによりメロディ生成モデルを学習する手法も提案されている [12]。しかし、この手法のように音符を単語に見立てることは、区切りの単位が細かすぎると考えられる。

なぜなら、音階の数は自然言語の語彙数に比べて極端に少なく、メタ情報を付加しても、単語に匹敵するような表現力を持たせることは困難であると考えられるからである。

Word2vecのフレームワークをメロディに応用する上で最も大きな問題はいかにしてメロディにおける「単語」を定義するかにある。メロディを構成する音符は、音名と音価という二つの要素を含んでおり、それぞれの要素を単語における文字に見立てることが可能であると考えられる。実際に、メロディにおいてよく使われる音名の数はひらがな五十音やアルファベット26文字と同じくらいのオーダーであり、その組み合わせにより単語が構成されると考えることは、音楽と自然言語との対応関係を考えて上でもリーズナブルなものであると考えられる。実際に、MML(Music Macro Language)などのように、音楽の要素を文字や記号で表すことでメロディを文章で記述するような表現方法が存在する。音符が文字、メロディが文章に対応すると仮定したとき、単語に対応する音楽的な概念はどのようなものであろうか? Bernsteinは講演の中で、自然言語における単語は音楽におけるフレーズに対応すると述べている [13], [14]。多くの文章において、重要な単語は繰り返し出現するものであり、メロディ中に繰り返し出現するフレーズに近いものであると考えられる。本稿では、メロディ内のフレーズが単語にあたるものと仮定して、メロディセグメンテーションによりフレーズを取得する。

メロディをフレーズに分割する手法はいくつか提案されており [15]、それらは大きく二種類のアプローチに分けられる。データドリブンのアプローチとモデルドリブンのアプローチである。データドリブンのアプローチでは、学習データに基づきセグメンテーション境界を決定する [16]。そのため、フレーズの境界は学習データの品質に影響を受けてしまい、データによってフレーズの定義に揺らぎが生じてしまうという問題が起こりえるため好ましくない。モデルドリブンのアプローチとしては、Lerdahlらが提案したGTTM[1]におけるGPR(Grouping Preference Rules)や、CambouropoulosによるLBDM(Local Boundary Detection Model) [17]、TemperleyによるGrouperなどが挙げられる [18]。それらの中でも、GPRは心理学的な実験によりその有効性が検証されている [19]。そこで、我々はGTTMにおけるGPRを用いてメロディのセグメンテーションを行うことで、word2vecのフレームワークをメロディへと拡張することを目指す。

## 3. データの準備

Word2vecのようにニューラルネットに基づくモデルを構築する上で、学習データの量はその精度に大きく影響する要素である。本研究では、楽曲内の音符の連なりによって表現されるメロディのみを学習の対象とする。このメロディの量は多ければ多いほど好ましい。Mikolovらに

よる word2vec の原論文 [4] では、学習データの量は 16 億単語にもものぼる。できる限り多くのデータを利用するために、我々はポピュラー音楽 176,581 曲分の MIDI データによって構成される the Lakh MIDI dataset[20] を利用した。著者らが知りうる限り、このデータセットはメロディの分析を行うことが可能な最大の楽曲データセットである。The Lakh dataset は Web 上からクロールされた MIDI ファイルによって構成されている。

これらの MIDI ファイルすべてからメロディ情報のみを抽出できれば、大規模な学習データが簡単に準備できる。MIDI ファイルは、各楽器の演奏情報がトラックに分かれているため、混合音を対象とした信号処理において必要となる音源分離等のような前処理を必要としない。しかし、Raffel らによると、MIDI ファイルのトラックの中のどのトラックがメロディパートに該当するのかを特定するための規則はない [21]。そのため、176,581 曲分の MIDI ファイルすべてに関して、どのトラックがメロディトラックであるかを明確に判定することはできない。特に、the Lakh dataset のように、Web を通じて収集した “wild” なデータセットの場合にはそれがさらに困難となる。MIDI ファイルには、各トラックの演奏楽器を指定するためのプログラムナンバーという情報があり、メロディトラックは “acoustic piano” や “synth voice” などのような楽器で演奏されることが多い。これらの楽器音を使用するトラックをメロディトラックであるとして、プログラムナンバーを用いたメロディ抽出を行うことも可能ではあるが、それらの楽器で演奏されたすべてのトラックがメロディにあたるわけではないため、学習データに多くのノイズを加えてしまうことになる。そのような不確定な要素によってメロディトラックを抽出するのではなく、我々は高い確率でメロディトラックであるといえるトラックのみを学習データに加えることとする。具体的には、トラックに付加されているメタデータとして、“melody” もしくは “vocal” の二つのキーワードのどちらかを文字列として含んでいるトラックのみをメロディトラックであるとして、抽出を行った。その結果、全 176,581 曲分のデータの中から、10,853 曲分のメロディデータを抽出することができた。

抽出した 10,853 曲分のメロディトラックは、すべて Web からクロールされた “wild” な MIDI ファイルから取得したデータであり、多くのノイズやタイミングの曖昧さ、音価の揺らぎが含まれている。そのため、メロディの表現が不明瞭なことが多い。特に、多くのケースで音符のオンセットの時間が正確であったとしても、オフセットのタイミングがデータによって異なるものとなっている。これらの理由から、MIDI イベントのオンセット、オフセットの間隔を単純に記録するだけではメロディを楽譜通りには取得できない。そこで、元の楽譜になるべく近いメロディ情報を得るために、八分休符よりも短い長さの休符について

は前に鳴った音の余韻であると判断し、前の音の長さに加えるという処理を行った。言い換えると、本手法では八分休符以上の長さの無音区間のみを休符として扱った。その結果、本手法によって得られたメロディには、MIDI における多様な演奏表現から生じる不自然な長さの休符があまり見られなくなった。

上記のような処理によって得られた 10,853 曲分のメロディに対して、次章で記述する前処理を適用することで melody2vec を実現する。

## 4. 前処理

メロディセグメンテーションや転調、オクターブの正規化からなる前処理によって Melody2vec モデルを学習するためのフレーズを取得する。データセットから抽出されたメロディは、GPR によってゲシュタルトに基づいてフレーズへと分割される。さらに、調が混在しているメロディデータに対して、C メジャーへの転調を行うことでフレーズが持つ意味を揃える。

### 4.1 GPR によるメロディフレーズの取得

GTTM による音符のグルーピングは認知心理学におけるゲシュタルトと呼ばれる概念に基づき行われる。ゲシュタルトは、意味を持たない構成要素を統合していくことで見つかる意味を持つ一つの塊である。自然言語において、個々の文字は意味を持たないが、文字の集合である単語は意味を持っている。同じように音楽においても、ゲシュタルトの原理に基づき GTTM によって音符をグルーピングすることで音楽的な単語を定義することはリーズナブルであると考えられる。

GTTM には二つのグルーピング規則がある。グルーピング構文規則 (Grouping Well-Formedness Rules: GWFR) とグルーピング選好規則 (Grouping Preference Rules: GPR) である。GWFR はグループを構成する上で必ず満たさなければならない規則であり、隣接する音符同士のみがグループを構成できるなどの規則を定めている。GPR は、グルーピング構造の境界としての好ましさを定める規則である。例えば、グループの境界は音符の長さや音量などが変化する箇所に存在する傾向にあるといった規則である。

exGTTM と呼ばれる理論は、GTTM 理論を計算機上で実行可能なように拡張した理論である [22]。exGTTM は GTTM の規則における曖昧さを排除したものであり、本研究では exGTTM を参照してフレーズを取得する。

exGTTM における GPR によると、四つの連続した音符 (それぞれ  $n_1, n_2, n_3, n_4$  とする) について考えたとき、以下の条件を満たす音符がグループの境界にあたるものとみなされる。

$$ioi_{n_2-n_3} > ioi_{n_1-n_2} \quad \text{and} \quad ioi_{n_2-n_3} > ioi_{n_3-n_4} \quad (1)$$

フレーズの境界



図 1 GPR に基づくメロディセグメンテーション

Fig. 1 Melody segmentation based on a GPR in a GTTM.

$ioi_{n-m}$  は  $n$  番目と  $m$  番目の音符の間のオンセット間の間隔 (inter-onset-interval) を表す。

いくつかのメロディのサンプルで実験を行った結果、我々は上記の条件を満たす位置においてメロディを分割することでフレーズを取得することとした。上記のグルーピング規則に基づきメロディセグメンテーションを行った結果を図 1 に示す。図 1 から、音符に大きな変化が生じた箇所ではメロディの分割が行われていることがわかる。仮にメロディを小節毎に分割した場合、このような小節を跨いだフレーズの取得はできない。

4.2 調とオクターブの正規化

まったく同じフレーズであったとしても、調が違った場合にはその役割は異なる。例えば、“G, A, B” といった音符の連なりは C メジャーにも D メジャーにも現れうるが、D メジャーにおける “G, A, B” は C メジャーにおける “F, G, A” と同じ役割となる。そこで、学習データのメロディの調をすべて一様に C メジャーへと転調させることで、学習データ内のフレーズの役割を統一する。

同じように、同一の調における “D4, D4, E4” と “D5, D5, E5” の違いはオクターブのみであり、コンテキストにもよるが、これらのフレーズの意味は同じであると考えられる。そこで、オクターブについても正規化を行い、中心となる音域がなるべく一致するようにする。

転調のアルゴリズムとして、メロディを構成する音名に関するヒストグラムに基づく転調手法を提案する。具体的には、音名のヒストグラムが C メジャーを表すテンプレート (C, D, E, F, G, A, B) に最も一致する方向に移調させることで転調を行う。Raffel らのサーベイによると、the Lakh dataset に含まれている楽曲の調の多数を占めるのはメジャー調である。そこで、すべての調をメジャーであると見なし、C メジャーへの転調を行う。そのため現状では、元々がマイナー調の曲については、C メジャーと同じ構成音を持つ平行調である A マイナーとの違いが区別できない。

楽曲の調を推定するための手法はいくつか提案されているが [23], [24], [25], 本手法による転調を行った結果で melody2vec の学習を行った場合が、セグメンテーション後のフレーズの語彙数が最も少ないという結果になった (表 1)。語彙数が少ないということは、フレーズの分散が抑えられているということに相当する。自然言語における単語の語彙数と比べ、メロディフレーズの語彙数は大きく

表 1 各転調手法による前処理を加えた結果の語彙数比較

Table 1 The number of vocabulary as a result of preprocessing by each method.

転調手法	フレーズの語彙数	
	オクターブ正規化なし	オクターブ正規化あり
転調前	318,783	306,192
SAM	320,726	306,324
提案手法	302,502	<b>286,003</b>

なる傾向にあるため、語彙数をなるべく抑えたフレーズの取得を実現する手法は、効率的なモデル構築をする上で望ましい。

我々は、提案転調手法を SAM (Spiral Array Model) に基づいて実装した転調手法と比較した [25]。図 2 は転調前、SAM による転調後、提案手法による転調後のそれぞれの音名のヒストグラムである。提案手法による転調を行った結果、最も多くの C メジャー構成音を含んでいることがわかる。そもそもの元データに多くのノイズを含んでいる今回のようなケースには、学習のための前処理としての転調には、提案手法のような単純な転調手法が適しているものと考えられる。

また、SAM では、 $\flat$  と  $\sharp$  の間に区別が必要だが、MIDI のデータからはそれらの区別ができないため、すべての  $\flat$  を  $\sharp$  として扱った。それも SAM によって高精度な転調ができなかった原因の一つであると考えられる。

上記の転調に加えて、オクターブに関する正規化も行った。各楽曲のメロディに含まれる音名のオクターブに関する頻度を数え、最頻オクターブが “C4” と同じオクターブとなるようにシフトする。オクターブ正規化を行う前後のノートナンバー (音名+オクターブ情報) に関するヒストグラムを図 3 に示す。オクターブ正規化によってヒストグラムの分散が少なくなっていることがわかる。

5. Melody2Vec

前処理の結果得られた 957,628 フレーズを学習データとして、melody2vec モデルを構築する。語彙数は 286,003 であった。Melody2vec モデルは、Mikolov らが word2vec の原論文 [4] で提案した skip-gram モデルを適用することで学習する。フレーズへと分割されたデータは、単語に分かち書きされた文章と同じステップで学習アルゴリズムを適用できる。様々なパラメータ、条件で実験を行った結果、学習するベクトルの次元数は 100、窓幅は 3 とした。

学習の後、得られたフレーズのベクトルを使って任意のフレーズに関する最近傍探索を行い、分散表現の品質をテストした。任意のフレーズをクエリとしたときの上位三近傍フレーズとそのときのフレーズ間のコサイン類似度を表 2 に示す。表中の “C4 : 1/4” という表現は四分音符の C4 を表す。

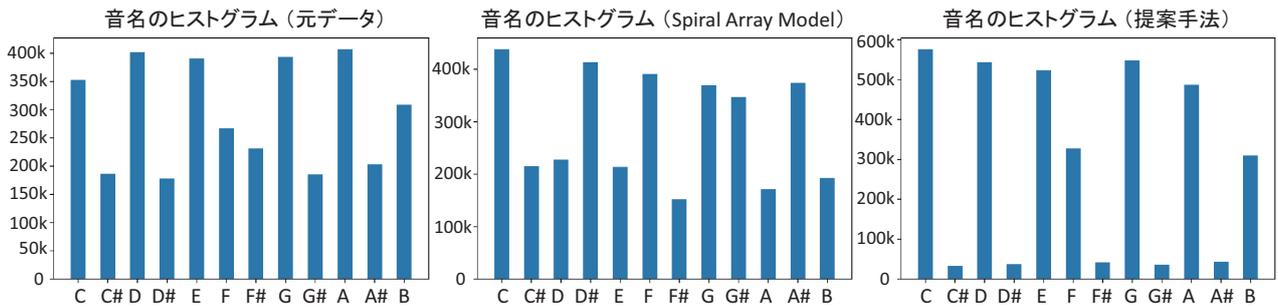


図 2 転調手法の違いによる音名ヒストグラムの比較. 転調なし (左), Spiral Array Model を用いた転調 (中央), 提案手法による転調 (右)

Fig. 2 Comparison of note name histogram without key modulation (left), key modulation with the Spiral Array Model (center), and key modulation with proposed method (right).

オクターブ正規化前後のノートナンバーの分布

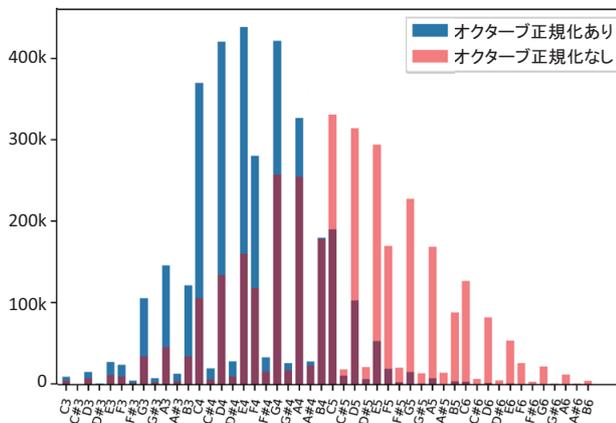


図 3 オクターブ正規化前後のノートナンバーの分布

Fig. 3 Histogram of note number before and after the octave normalization.

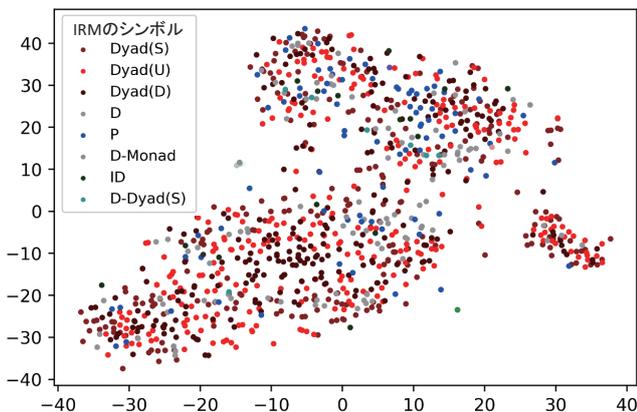


図 4 t-SNE による二次元平面へのマッピング

Fig. 4 2D embedding visualization by t-SNE.

表 2 クエリに対して類似度が高い上位三近傍フレーズ

Table 2 Examples of top three phrases with similarity to the query.

クエリ 1	D4:1/16-G4:1/16-G4:1/16- D4:1/16-G4:1/8	類似度
No.1	D4:1/16-G4:1/8-D4:1/16-R:4/4	0.711
No.2	A4:1/16-B4:1/8-G4:1/8-R:1/8	0.692
No.3	D4:1/16-G4:1/8	0.655
クエリ 2	G3:1/8-C4:1/8-D4:1/8- C4:1/8-D4:1/4	類似度
No.1	G3:1/8-C4:1/8-D4:1/8- C4:1/8-D4:1/6-R:4/4	0.858
No.2	D4:1/8-D4:1/8-G4:1/6- F4:1/6-D4:1/4	0.761
No.3	D4:1/8-G3:1/8	0.679
クエリ 3	C4:1/6-R:3/8-E4:1/8-D4:1/8- D4:1/8-D4:1/8-C4:1/6-R:1/4	類似度
No.1	C4:1/6-R:3/8-E4:1/8-D4:1/8- D4:1/8-D4:1/8-C4:1/16- A3:1/16-C4:1/12-R:1/4	0.999
No.2	C4:1/8-D4:1/8-E4:1/4-E4:1/4- E4:1/8-D4:1/4-R:1.5	0.752
No.3	E4:1/8-D4:1/8-D4:1/8-D4:1/8- C4:1/6-R:1/4	0.745

## 6. 結果

### 6.1 結果の可視化

#### 6.1.1 t-SNE による二次元平面へのマッピング

フレーズに関する 100 次元のベクトルを t-SNE[26] によって二次元平面にマッピングして可視化した. 図 4 は, 学習データにおける出現頻度上位 1000 フレーズを t-SNE によりプロットしたものである. 各プロットがフレーズを表しているが, 図の視認性を落とさないために, プロット

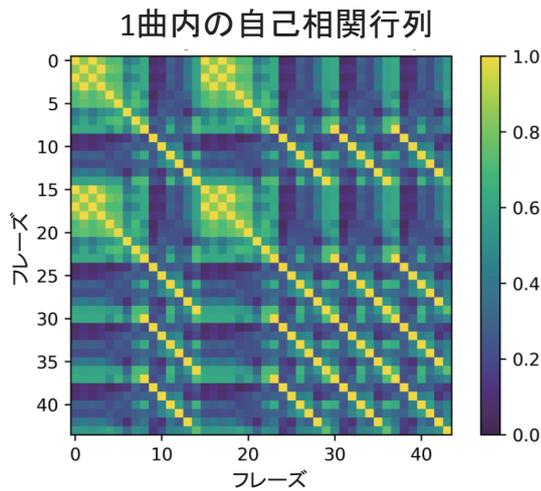


図 5 1 曲内のフレーズの自己類似度行列

Fig. 5 Self-similarity matrix of a musical piece.

にラベルは付加しなかった。ラベルを付加する代わりに、IRM (暗意実現モデル) [2], [3] に基づいて与えられるシンボルに応じてプロットを色付けした。1000 のフレーズには、19 種類の多様なシンボルが付与されたが、図 4 にはその中でも出現頻度が高かったシンボルに関してのみ凡例を示している。IRM によると、同じシンボルが与えられるフレーズは似ているフレーズであると見なすことができる。二音のみからなるフレーズに与えられるシンボルである dyad は IRM の原論文では一種類のみしか存在しないが、頻出フレーズには、二音のみからなるフレーズが多く出現する都合で、二音間の音程に基づき dyad を三つに分類した。Dyad (S) は二音が同じ高さである場合、dyad (U) は二音目が一音目よりも高い場合、dyad (L) は二音目が一音目よりも低い場合に付与した独自のシンボルである。その他のシンボルについては、元々の IRM と同じシンボルを使用している。音符数が多いフレーズに対しては、フレーズの先頭から複数のシンボルを付与している。

1000 フレーズ分のプロットに対する色付けは、色ごとにクラスタリングできるようなグループにはなっていないが、近いプロット同士が同じ色になっているケースが多いことが確認できる。実際に、最近傍プロット同士が同じ色である割合は 37.6% であった。三近傍プロット内に同じ色のプロットがある割合は 68.8% となり、概ね近いプロットの色が同じ色となっていることがわかる。図 4 にプロットした 1000 のフレーズに対して 19 種類の IRM のシンボルが付与されたことを考えると、この数字は十分に高いことがわかる。

### 6.1.2 曲内のフレーズ間類似度の可視化

一曲の中で、フレーズベクトルに基づく類似度がどのように変化していくかを検証した。学習データ内に存在する一曲の楽曲選び、そのメロディを構成するフレーズのベクトルを取得した。取得した一曲内のベクトル同士のコサイ

表 3 評価実験で用いられた各試行のフレーズ A, B における提案類似度と Levenshtein 距離

Table 3 The proposed similarity and Levenshtein distance of phrase set used in the user study.

試行	提案類似度		Levenshtein 距離	
	A	B	A	B
1	0.903	$2.160 \times 10^{-6}$	10	4
2	0.917	$2.270 \times 10^{-6}$	12	9
3	0.999	$8.703 \times 10^{-2}$	5	4
4	0.911	$1.740 \times 10^{-6}$	10	7
5	0.901	$1.080 \times 10^{-7}$	6	5
6	0.901	$2.040 \times 10^{-6}$	9	8
7	0.904	$9.570 \times 10^{-7}$	8	6
8	0.916	$4.430 \times 10^{-7}$	12	6
9	0.985	$1.860 \times 10^{-8}$	10	7
10	0.901	$1.080 \times 10^{-7}$	6	5
平均	0.924	$8.704 \times 10^{-3}$	8.8	6.1

ン類似度を計算することで求めた自己類似度行列を図 5 に示す。楽曲は Green Day の “Good Riddance” であり、その楽曲の構造は「A メロ→B メロ→サビ→B メロ→サビ (三回)」となっている。図の黄色い箇所は特に類似度が高い領域を表しており、曲の構造を反映している結果となっている。特に、この曲において A メロと B メロは違うけれども似ているメロディで構成されており、A メロ同士、B メロ同士の類似度に比べて若干類似度は低いものの高い類似度を示している。この結果から、本手法により獲得できたフレーズベクトルは、楽曲の構造分析にも応用できる可能性が示唆された。

## 6.2 主観評価実験

本手法によって得られたフレーズベクトルの有効性を検証するために主観評価実験を行った。ベクトル間の類似度について XAB 法によって評価することで、間接的にベクトルの有効性を検証した。

評価実験には 17 名の男女が参加者した。参加者は、一回の試行で三種類のメロディフレーズ (それぞれ Q, A, B とする) を聴取し、それを 10 回繰り返す。Q のフレーズはクエリであり、A は提案ベクトルを用いたコサイン類似度が最も高いフレーズ、B はコサイン類似度が最も低い Levenshtein 距離 (編集距離) が A に比べて小さいフレーズである。参加者は試行毎に Q に似ているフレーズが A と B のどちらであるかを回答した。試行は計 10 セット行われ、A と B を聴く順番を試行毎に入れ替えることで、聴取順序による影響を抑えた。

表 3 に本評価実験で使用した全 QAB セットについての、提案手法によって求めたコサイン類似度 (以下、提案類似度とする) と Levenshtein 距離を示す。クエリには、学習データにおいて 10 回以上出現したフレーズの中からランダムに 10 のフレーズが選ばれた。これらの 10 のフレーズ

表 4 提案類似度の有効性を検証する主観評価実験の結果。

Table 4 The result of the user study which verifies the effectiveness of the proposed similarity measure.

試行	提案手法	Levenshtein 距離
1	17	0
2	16	1
3	15	2
4	13	4
5	17	0
6	15	2
7	12	5
8	17	0
9	15	2
10	15	2

において、最近傍のベクトルとのコサイン類似度が 0.9 に満たないようなフレーズについては再度ランダムに選びなおすことで、すべての Q-A の提案類似度が 0.9 以上となるように設定した。これは、十分に似ているフレーズがあることが保証されたセットで評価を行うためである。表 3 に示すように、A についての提案類似度はすべて 0.9 以上である。また、提案類似度が最も低いフレーズ B を求めた際に、Levenshtein 距離が Q-A 間の距離よりも大きい場合には、次に類似度が低いフレーズを B とし、Q-A に比べて Levenshtein 距離が小さいフレーズが見つかるまで探索を行うことでフレーズ B を決定した。表 3 に示すように、Q-B の Levenshtein 距離は必ず Q-A の距離よりも小さくなっている。Levenshtein 距離の計算において、音名と音価はそれぞれ別の物として扱った。例えば、フレーズ内の一つの音符が “C4:1/4” から “C4:1/8” に変化している場合の Levenshtein 距離は 1 であるとし、もしも “D4:1/8,” に変化している場合は Levenshtein 距離を 2 とした。

主観評価実験の結果を表 4 に示す。表中の数値は、クエリに対してより似ているフレーズとして選ばれた回数を表しており、その最大値は 17 である。表からわかるように、提案類似度によるフレーズ (A) を選択した参加者の数のほうが B を選んだ参加者よりも多く、全回答のうちの 89.4% が提案類似度によって得られたフレーズであった。

Levenshtein 距離が小さいフレーズは、共通する音符を多く保持するものであるため、必然的に似たメロディとなるものである。例えば、“C4:1/4-D4:1/4-E4:1/4” と “C4:1/4-D4:1/4-G4:1/4” の間の Levenshtein 距離は 1 であり、最後の音以外はすべて共通の音符である。一方で、提案手法によるフレーズ A は、フレーズ B に比べて Levenshtein 距離が大きいものであった。つまり、共通点が少ないけれども似ているフレーズを見つけることができていることを示している。この結果から、本手法によって得られたベクトルは、共通点が少ない違うフレーズ同士であって

元のメロディ



置換後のメロディ



図 6 置換前後のメロディの比較

Fig. 6 Musical score before and after the melody replacement.

も似ている場合には近くに配置されるという特徴を持っているということが期待できる。

メロディの類似度についての ground truth となるデータは存在しないため、提案手法によって意味的な類似度が十分に記述できているかどうかはまだ検討の余地がある。これについては、より踏み込んだ検証が必要となると考えており、今後の研究で検証していきたい。

7. メロディフレーズの置換

Melody2vec モデルの応用例として、メロディ内のフレーズの置換に関する例を示す。5章の表 2 に示したように、曲中の任意のフレーズをクエリとすることで、該当フレーズに似ている別のフレーズを検索して置き換えることが可能である。

ここで、音楽において音の長さは重要な要素である。そのため、置換前後のフレーズの長さは完全に一致している必要がある。そこで、任意の置き換え対象フレーズを基に類似フレーズを検索する際に、置き換え候補となるフレーズは必ず同一の長さとなるような制約を設けた。つまり、置き換え候補となるフレーズは、置き換え対象となるフレーズと同一の長さのフレーズの中で最も類似度が高いフレーズとなる。

図 6 にメロディフレーズの置換の例を示す。図中の楽譜はメロディフレーズの置換前後の楽譜である。楽曲は The Beatles の “All my loving” であり、冒頭の 1 フレーズを置き換えの対象として選んだ。この曲は、学習データにも含まれている楽曲である。

フレーズの置換が自然に行われているかを検証するために主観評価実験を行った。17名の参加者が二つのメロディ (それぞれメロディ A, メロディ B とする) を聴取した。二つのメロディは、どちらもフレーズの置き換えが行われた後のメロディである。参加者には、メロディが一箇所編集されている旨を伝えた上でメロディを聴いてもらい、それぞれどこが編集されている箇所であるかを回答してもらった。回答は、秒を単位として行ってもらい、前後 1 秒以内にフレーズ置換箇所があった場合に正解であるとした。

メロディ A は図 6 に示した置換後のメロディであり、図中に示されていない箇所も含めて、再生時間は約 24 秒であった。メロディ B は学習データに含まれていないメロ

表 5 フレーズ置換箇所に関する正答率

Table 5 Percentage of participants who could point out the replacement part of the melody.

	メロディ A	メロディ B
曲を聴いたことがある人の正答率	66.7%	50.0%
曲を聴いたことがない人の正答率	14.3%	0.0%
全体の正答率	23.5%	25.0%

ディで、レミオロメンの“3月9日”という曲である。メロディ B の長さは約 37 秒であった。参加者には、編集箇所がどこであるかを回答してもらうと共に、その曲を聴いたことがあるかどうかについても回答してもらった。

この評価実験の結果を表 5 に示す。楽曲を聴いたことがある人の正答率は、聴いたことがない人の正答率に比べて高かった。また、楽曲を聴いたことがない人の正答率は低く、メロディ A を聴いたことがない 14 名のうち 2 名が正答しただけであり、メロディ B については正答者がいなかった。この結果は、フレーズの置換が自然に行われており、メロディを聴いたことがない人にとっては本物のメロディよりも違和感がないものとなっていたことを示している。メロディを聴いたことがある人の置換箇所についての正答率が比較的高かったが、それは置換の不自然さを示すわけではなく、知っている人にとっては違うフレーズに置き換わっていることがわかる程度の置換が行えているということを示すものである。

今後の研究において、フレーズ置換を基にしたメロディの編集インタフェースの実装を計画している。メロディ内のユーザが変更したい箇所を指定することで、置き換え候補となるフレーズを提示し、インタラクティブにメロディを編集できるようなシステムの実現を目指している。

## 8. まとめ

本稿では、word2vec のフレームワークを音楽のメロディに拡張した melody2vec を提案した。GTTM における GPR を用いたメロディセグメンテーション手法により、word2vec のフレームワークのメロディへの拡張が実現した。1 万を超えるメロディデータを用いて melody2vec のモデルを構築し、主観評価実験を通じてその有効性を示した。さらに、melody2vec モデルを応用したアプリケーションとして、メロディ内のフレーズ置換の例を紹介した。

### 8.1 Limitations

本手法にはいくつかの課題がある。一つは、現在の前処理では、転調の際に C メジャーと A マイナーを区別することができないことである。モデルの有効性をさらに向上させるために、調の推定をより正確に行う必要がある。ま

た、現在使用している 10,853 曲分のメロディデータは the Lakh dataset の全楽曲のうちの 6% にすぎず、現状では大規模データの恩恵を十分に受けているとは言いがたい。より多くのデータを学習データとして使用可能とするためには、MIDI ファイル内のメロディトラック同定手法 [27] などを利用することが有効であると考えている。

現在、提案モデルはメロディ内のフレーズに関するモデルにとどまっている。実際にメロディ全体のベクトル表現を獲得するには、さらに後処理を加える必要があると考えている。例えば、word2vec の文書への拡張である doc2vec [28] のアプローチが有効かもしれない。

## 8.2 議論

Word2vec の原論文 [4] において最もインパクトが強かった結果であるベクトルの演算について、本研究ではその効果を検証することができなかった。ベクトル演算の実装は容易にできたのであるが、その結果の良し悪しを検証することが叶わなかった。なぜならば、我々は複数のメロディフレーズを足したらどのようなメロディになるかなどといったことを想像することができないからである。ベクトルの演算については今後の課題として研究を続けていくつもりである。

多くの楽曲にはメロディに合った歌詞がついている。歌詞は単語の集合であり、メロディに関するベクトルと同時に単語のベクトルも取得することができる。このような複数要素の組み合わせによるベクトル表現は、音楽検索の新たな可能性に繋がるのではないかと期待している。メロディフレーズの分散表現は、音楽情報検索分野における重要な要素となりうるものである。検索だけでなく編集等にも応用可能な技術であり、将来の音楽情報処理研究にもたらす可能性は非常に大きいものと考えている。今後もメロディの分散表現とその応用に関する研究を続けていきたい。

謝辞 本研究結果の二次元可視化にあたり、NTT メディアインテリジェンス研究所の矢澤櫻子氏に、IRM (暗意実現モデル) に関する助言を頂いたことを感謝する。

## 参考文献

- [1] Lerdahl, F. and Jackendoff, R. S.: *A Generative Theory of Tonal Music*, The MIT press (1983).
- [2] Narmour, E.: *The Analysis and Cognition of Basic Melodic Structures*, The University of Chicago Press (1990).
- [3] Narmour, E.: *The Analysis and Cognition of Melodic Complexity*, The University of Chicago Press (1992).
- [4] 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).
- [5] Volk, A., Van Kranenburg, P., Garbers, J., Wiering, F., Veltkamp, R. C. and Grijp, L. P.: *A Manual Annota-*

- tion Method for Melodic Similarity and the Study of Melody Feature Sets., *Proceedings of the International Symposium on Music Information Retrieval*, pp. 101–106 (2008).
- [6] Hamanaka, M., Hirata, K. and Tojo, S.: Melody Morphing Method Based on GTTM., *Proceedings of the International Computer Music Conference*, pp. 155–158 (2008).
- [7] Hirata, K. and Matsuda, S.: Interactive Music Summarization based on Generative Theory of Tonal Music, *Journal of New Music Research*, Vol. 32, No. 2, pp. 165–177 (2003).
- [8] Saito, M. and Matsui, Y.: Illustration2Vec: A Semantic Vector Representation of Illustrations, *SIGGRAPH Asia 2015 Technical Briefs*, SA '15, New York, NY, USA, ACM, pp. 5:1–5:4 (online), DOI: 10.1145/2820903.2820907 (2015).
- [9] Vosoughi, S., Vijayaraghavan, P. and Roy, D.: Tweet2Vec: Learning Tweet Embeddings Using Character-level CNN-LSTM Encoder-Decoder, *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1041–1044 (2016).
- [10] Wang, D., Deng, S. and Xu, G.: Sequence-based Context-aware Music Recommendation, *Information Retrieval Journal*, pp. 1–23 (2017).
- [11] Herremans, D. and Chuan, C.-H.: Modeling Musical Context with Word2vec, *Proceedings of the First International Conference on Deep Learning and Music*, pp. 11–18 (2017).
- [12] Shin, A., Crestel, L., Kato, H., Saito, K., Ohnishi, K., Yamaguchi, M., Nakawaki, M., Ushiku, Y. and Harada, T.: Melody Generation for Pop Music via Word Representation of Musical Properties, *arXiv preprint arXiv:1710.11549* (2017).
- [13] Bernstein, L.: Lecture II, Musical Syntax, in “Unanswered Question” (1973).
- [14] 平田圭二, 東条敏: バーンスタインの「答えのない質問」再考: 計算論的音楽の理論の枠組みについて, 人工知能学会全国大会論文集, Vol. 28, pp. 1–4 (2014).
- [15] López, M. R. and Volk, A.: Automatic Segmentation of Symbolic Music Encodings: A Survey (2012).
- [16] Bod, R.: Memory-based Models of Melodic Analysis: Challenging the Gestalt Principles, *Journal of New Music Research*, Vol. 31, No. 1, pp. 27–36 (2002).
- [17] Cambouropoulos, E.: The Local Boundary Detection Model (LBDM) and its Application in the Study of Expressive Timing., *ICMC* (2001).
- [18] Temperley, D.: *The Cognition of basic Musical Structures*, MIT press (2004).
- [19] Deliege, I.: Grouping Conditions in Listening to Music: An Approach to Lerdahl & Jackendoff’s Grouping Reference Rules, *Music Perception: An Interdisciplinary Journal*, Vol. 4, No. 4, pp. 325–359 (1987).
- [20] Raffel, C.: *Learning-based Methods for Comparing Sequences, with Applications to Audio-to-midi Alignment and Matching*, Columbia University (2016).
- [21] Raffel, C. and Ellis, D. P.: Extracting Ground-Truth Information from MIDI Files: A MIDIfesto., *Proceedings of the International Symposium on Music Information Retrieval*, pp. 796–802 (2016).
- [22] Hamanaka, M., Hirata, K. and Tojo, S.: ATTA: Implementing GTTM on a Computer., *Proceedings of the International Symposium on Music Information Retrieval*, pp. 285–286 (2007).
- [23] Longuet-Higgins, H. C. and Steedman, M. J.: On Interpreting Bach, *Machine intelligence*, Vol. 6, pp. 221–241 (1971).
- [24] Castellano, M. A., Bharucha, J. J. and Krumhansl, C. L.: Tonal Hierarchies in the Music of North India., *Journal of Experimental Psychology: General*, Vol. 113, No. 3, pp. 394–412 (1984).
- [25] Chew, E.: *Towards a mathematical model of tonality*, Massachusetts Institute of Technology (2000).
- [26] 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).
- [27] Rizo, D., Leon, P., Perez-Sancho, C., Pertusa, A. and Inesta, J.: A Pattern Recognition Approach for Melody Track Selection in MIDI Files., *Proceedings of the International Symposium on Music Information Retrieval*, pp. 61–66 (2006).
- [28] Le, Q. and Mikolov, T.: Distributed Representations of Sentences and Documents, *Proceedings of the International Conference on Machine Learning*, pp. 1188–1196 (2014).