

# 近親調を用いた楽曲クラスタリングシステムの構築に向けて

長澤 槇子<sup>†</sup> 渡辺 知恵美<sup>‡</sup> 伊藤 貴之<sup>‡</sup> 増永 良文<sup>‡</sup>

<sup>†</sup>お茶の水女子大学大学院 人間文化研究科博士前期課程 数理・情報科学専攻

<sup>‡</sup>お茶の水女子大学 理学部情報科学科

〒112-8610 東京都文京区大塚 2-1-1

E-mail: <sup>†</sup> makiko@db.is.ocha.ac.jp, <sup>‡</sup> {chiemi, itot, masunaga}@is.ocha.ac.jp

**あらまし** 楽曲には様々な音楽的特徴があり、その中のコード進行は楽曲のベースとなるもので不変性が高いものである。そこで本研究では、コード進行に着目した楽曲クラスタリングシステムを提案する。コード進行の類似度に関して、我々は音楽理論にて定義されている近親調をもとに定義した。本システムにより、「作曲者とよく使うコード進行パターンとの関係」や「異なる楽曲間で類似したコード進行パターンを用いている」といった情報の抽出が可能となる。また、クラスタリング結果をもとに、楽曲名、作曲者、発売日、その楽曲のヒットチャートといった属性を追加し、「ヒット曲とコード進行の関係」や「年代とコード進行の関係」といった他属性との相関関係をマイニングすることを目指す。

**キーワード** マルチメディア処理, マイニング, 音楽情報処理, クラスタリング, 近親調

## Towards a Construction of a Music Clustering System using Relative Key

Makiko NAGASAWA<sup>†</sup> Chiemi WATANABE<sup>‡</sup> Takayuki ITO<sup>‡</sup> and Yoshifumi MASUNAGA<sup>‡</sup>

<sup>†</sup> Graduate Division of Mathematics and Computer Science (Master's Program), Ochanomizu University

<sup>‡</sup> Department of Information Science, Ochanomizu University

2-1-1 Otsuka, Bunkyo-ku, Tokyo, 112-8610 Japan

E-mail: <sup>†</sup> makiko@db.is.ocha.ac.jp, <sup>‡</sup> {chiemi, itot, masunaga}@is.ocha.ac.jp

**Abstract** There are various features in music. Here the chord progression is the most invariant feature and the basis of music. In this paper we propose a music clustering system focusing on the chord progression. We define similarity of chord progression according to relative keys defined by music theory. We believe that this system can extract the information such as chord progression patterns often used by the same composer, or chord progression patterns used by different music. Our future target is the mining of relativity among various attributes of music, such as titles, composers, date of release, and ranking of hit charts, based on the clustering results by this system.

**Keyword** Multimedia processing, Mining, Music information processing, Clustering, Relative key

### 1. はじめに

世の中にはクラシック, ポップス, ジャズなど様々な楽曲が存在し, これらの楽曲には様々な音楽的特徴がある. 例えば以下のようなものが挙げられる.

- (1) 楽器の種類
- (2) ハーモニーの変化
- (3) コード進行
- (4) リズムパターン

### (5) モチーフ

(1)では, 楽曲の楽器の種類を判別することが可能である. (2)では, 作曲者により特定のハーモニーパターンが頻出する場合がある. (3)では, 楽曲の年代や歌手により特定のコード進行パターンが頻出する場合がある. (4)では, 音楽ジャンルによってリズムパターンが決まっており, 音楽ジャンルを特定することが可能である. (5)のモチーフは, 音楽では動機とも呼ばれており, いくつかの音符ないしは, 休符の特徴的な連なり

のことであり、楽曲の中で繰り返し用いられる。

これらの中で(3)のコード進行は、楽曲の旋律に伴う和音(コード)の変化のことである。音楽ではコード進行がベースとなっており、リズムパターンやメロディに比べ不変性の高いものである。よって、楽曲の特徴要素としてコード進行は重要なものと考え、本研究では、(3)に挙げたコード進行に着目し楽曲のクラスタリングを行う。コード進行の類似度に関して、我々は音楽理論にて定義されている近親調をもとに定義した。近親調に関しては、第 4.4.1 節にて述べる。

本システムが実現すれば、「作曲者がよく使うコード進行パターン」や「異なる楽曲間で類似したコード進行パターンを用いている」といった情報を抽出することが可能となる。また、クラスタリングの結果をもとに、楽曲名、作曲者、発売日、その楽曲のヒットチャートといった属性を追加し、「ヒット曲とコード進行との関係」や「作曲者とコード進行の関係」といった、他属性との相関関係を抽出することが可能となる。

## 2. 関連研究

楽曲クラスタリングシステムに関する研究について、松田ら[1]は、テキストマイニング技術を音楽情報へ適用し音楽解析・音楽分類について提案した。音やコードに適切なテキストを割り当て、メロディとコードについての分析を行うというものである。しかし、クラスタリングをする際に近親調を考慮していないため、コード進行理論の観点から類似したものが同じクラスタに含まれているかは明確ではない。梶[2]は、楽曲に対する多様な解釈を Web 上から収集する音楽アノテーションシステムを構築し、楽曲検索システムとプレイリスト推薦システムを構築した。楽曲検索システムでは、タイトルや作詞・作曲者、アノテーションを用いたキーワード検索、コード進行による検索、曲の構成による絞込み検索が可能である。コード進行による検索に関してはコードに数値を割り当て、各音楽コンテンツに対してコードのアノテーションを取得し曲ごとにコード進行を作成し、DP マッチングを行い検索要求との距離を測るというものである。吉原ら[3]は、旋律に対する和声付けに焦点を絞り、大量の雑多なデータから優位な情報を取り出し、取り出した情報を利用した半自動和声付けシステムを構築した。その際、様々なタイプの和声付けの実現のためにあらかじめ進行に特徴のある楽曲同士のクラスタリングを行っている。[2]、[3]は、クラスタリングを行う際に、近親調については考慮されていない。

そこで我々は近親調を考慮しコード進行に関しての距離を定義し、クラスタリングを行う。本論文では、コード進行のみに着目しクラスタリングを行ったが、

今後、メロディやリズムパターンといった要素についてもクラスタリングを行う予定である。

## 3. 楽曲クラスタリングの概要

本章では、近親調を用いた楽曲クラスタリングにおけるクラスタリング処理の概要について述べる。全体の処理の流れは以下の通りである。図 1 に処理全体の流れを示す。

### (1) コード進行の数値化

クラスタリングを行うにあたり、コード進行データを数値化する。数値化の手法については第 4.1 節にて述べる。

### (2) コード進行の正規化

コード進行の類似パターンを抽出するにあたり、コード進行を正規化する必要がある。正規化については、第 4.2 節にて述べる。

### (3) 楽曲の分割

本システムでは、楽曲の部分要素をクラスタリングしていくため、楽曲を分割する必要がある。楽曲の分割については、第 4.3 節にて述べる。

### (4) 距離の計算

分割した要素ごとの類似度を測るために、第 4 章で述べる定義に従って距離の計算を行う。

### (5) 類似パターン抽出

手順(4)にて求めた距離が近いものを類似しているとみなし類似パターンを抽出する。

### (6) クラスタリング

手順(5)にて抽出した類似パターンのクラスタリングを行う。第 5.1 節にて、クラスタリング手法について述べる。

### (7) 他属性との相関抽出

クラスタリング結果をもとに、楽曲名、作曲者、発売日、その楽曲のヒットチャートといった属性を追加し、「ヒット曲とコード進行との関係」や、「作曲者とコード進行との関係」といった、他属性との相関関係をマイニングする。

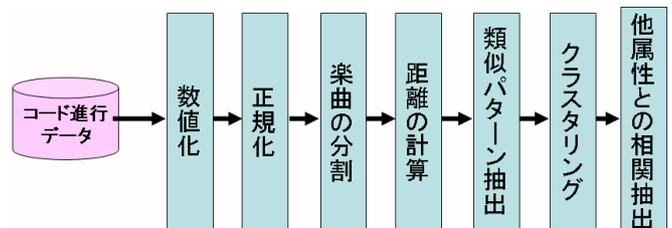


図 1 楽曲クラスタリングの流れ

## 4. コード進行に関する諸定義

### 4.1. コード進行の数値化

コード進行一覧を図 2 に示す[4]、本システムでは、

図中の第一行目の「 $C, C^\sharp, D^\flat, \dots, B$ 」をトーン、第一列目の「 $M, m, \dots, mM7$ 」をコードと名づける。トーンは図2で左から右に0, ..., 11, コードは図2で上から下に0, ..., 13と数値を割り当てる。また、音楽は時系列データであるので時間の情報を考慮する必要がある。本システムでは、四分音符を1拍として、一拍目を0としてコード進行を以下のように定義する。

【定義1】コード進行 Chord

$Chord = (トーン, コード, 時間)$

例えば、一拍目が  $D^\flat$  であった場合、 $D^\flat = (1, 0, 0)$ 、四拍目が  $F_m$  であった場合、 $F_m = (5, 1, 3)$ となる。

	$C$	$C^\sharp, D^\flat$	$D$	$D^\sharp, E^\flat$	$E$	$F$	$F^\sharp, G^\flat$	$G$	$G^\sharp, A^\flat$	$A$	$A^\sharp, B^\flat$	$B$
$M$	$C$	$C^\sharp, D^\flat$	$D$	$D^\sharp, E^\flat$	$E$	$F$	$F^\sharp, G^\flat$	$G$	$G^\sharp, A^\flat$	$A$	$A^\sharp, B^\flat$	$B$
$m$	$C_m$	$C^\sharp_m, D^\flat_m$	$D_m$	$D^\sharp_m, E^\flat_m$	$E_m$	$F_m$	$F^\sharp_m, G^\flat_m$	$G_m$	$G^\sharp_m, A^\flat_m$	$A_m$	$A^\sharp_m, B^\flat_m$	$B_m$
7	$C_7$	$C^\sharp_7, D^\flat_7$	$D_7$	$D^\sharp_7, E^\flat_7$	$E_7$	$F_7$	$F^\sharp_7, G^\flat_7$	$G_7$	$G^\sharp_7, A^\flat_7$	$A_7$	$A^\sharp_7, B^\flat_7$	$B_7$
$M7$	$C_{M7}$	$C^\sharp_{M7}, D^\flat_{M7}$	$D_{M7}$	$D^\sharp_{M7}, E^\flat_{M7}$	$E_{M7}$	$F_{M7}$	$F^\sharp_{M7}, G^\flat_{M7}$	$G_{M7}$	$G^\sharp_{M7}, A^\flat_{M7}$	$A_{M7}$	$A^\sharp_{M7}, B^\flat_{M7}$	$B_{M7}$
$m7$	$C_{m7}$	$C^\sharp_{m7}, D^\flat_{m7}$	$D_{m7}$	$D^\sharp_{m7}, E^\flat_{m7}$	$E_{m7}$	$F_{m7}$	$F^\sharp_{m7}, G^\flat_{m7}$	$G_{m7}$	$G^\sharp_{m7}, A^\flat_{m7}$	$A_{m7}$	$A^\sharp_{m7}, B^\flat_{m7}$	$B_{m7}$
$dim$	$C_{dim}$	$C^\sharp_{dim}, D^\flat_{dim}$	$D_{dim}$	$D^\sharp_{dim}, E^\flat_{dim}$	$E_{dim}$	$F_{dim}$	$F^\sharp_{dim}, G^\flat_{dim}$	$G_{dim}$	$G^\sharp_{dim}, A^\flat_{dim}$	$A_{dim}$	$A^\sharp_{dim}, B^\flat_{dim}$	$B_{dim}$
$m7^{-5}$	$C_{m7^{-5}}$	$C^\sharp_{m7^{-5}}, D^\flat_{m7^{-5}}$	$D_{m7^{-5}}$	$D^\sharp_{m7^{-5}}, E^\flat_{m7^{-5}}$	$E_{m7^{-5}}$	$F_{m7^{-5}}$	$F^\sharp_{m7^{-5}}, G^\flat_{m7^{-5}}$	$G_{m7^{-5}}$	$G^\sharp_{m7^{-5}}, A^\flat_{m7^{-5}}$	$A_{m7^{-5}}$	$A^\sharp_{m7^{-5}}, B^\flat_{m7^{-5}}$	$B_{m7^{-5}}$
$aug$	$C_{aug}$	$C^\sharp_{aug}, D^\flat_{aug}$	$D_{aug}$	$D^\sharp_{aug}, E^\flat_{aug}$	$E_{aug}$	$F_{aug}$	$F^\sharp_{aug}, G^\flat_{aug}$	$G_{aug}$	$G^\sharp_{aug}, A^\flat_{aug}$	$A_{aug}$	$A^\sharp_{aug}, B^\flat_{aug}$	$B_{aug}$
$sus4$	$C_{sus4}$	$C^\sharp_{sus4}, D^\flat_{sus4}$	$D_{sus4}$	$D^\sharp_{sus4}, E^\flat_{sus4}$	$E_{sus4}$	$F_{sus4}$	$F^\sharp_{sus4}, G^\flat_{sus4}$	$G_{sus4}$	$G^\sharp_{sus4}, A^\flat_{sus4}$	$A_{sus4}$	$A^\sharp_{sus4}, B^\flat_{sus4}$	$B_{sus4}$
$7sus4$	$C_{7sus4}$	$C^\sharp_{7sus4}, D^\flat_{7sus4}$	$D_{7sus4}$	$D^\sharp_{7sus4}, E^\flat_{7sus4}$	$E_{7sus4}$	$F_{7sus4}$	$F^\sharp_{7sus4}, G^\flat_{7sus4}$	$G_{7sus4}$	$G^\sharp_{7sus4}, A^\flat_{7sus4}$	$A_{7sus4}$	$A^\sharp_{7sus4}, B^\flat_{7sus4}$	$B_{7sus4}$
6	$C_6$	$C^\sharp_6, D^\flat_6$	$D_6$	$D^\sharp_6, E^\flat_6$	$E_6$	$F_6$	$F^\sharp_6, G^\flat_6$	$G_6$	$G^\sharp_6, A^\flat_6$	$A_6$	$A^\sharp_6, B^\flat_6$	$B_6$
$add9$	$C_{add9}$	$C^\sharp_{add9}, D^\flat_{add9}$	$D_{add9}$	$D^\sharp_{add9}, E^\flat_{add9}$	$E_{add9}$	$F_{add9}$	$F^\sharp_{add9}, G^\flat_{add9}$	$G_{add9}$	$G^\sharp_{add9}, A^\flat_{add9}$	$A_{add9}$	$A^\sharp_{add9}, B^\flat_{add9}$	$B_{add9}$
$m6$	$C_{m6}$	$C^\sharp_{m6}, D^\flat_{m6}$	$D_{m6}$	$D^\sharp_{m6}, E^\flat_{m6}$	$E_{m6}$	$F_{m6}$	$F^\sharp_{m6}, G^\flat_{m6}$	$G_{m6}$	$G^\sharp_{m6}, A^\flat_{m6}$	$A_{m6}$	$A^\sharp_{m6}, B^\flat_{m6}$	$B_{m6}$
$mM7$	$C_{mM7}$	$C^\sharp_{mM7}, D^\flat_{mM7}$	$D_{mM7}$	$D^\sharp_{mM7}, E^\flat_{mM7}$	$E_{mM7}$	$F_{mM7}$	$F^\sharp_{mM7}, G^\flat_{mM7}$	$G_{mM7}$	$G^\sharp_{mM7}, A^\flat_{mM7}$	$A_{mM7}$	$A^\sharp_{mM7}, B^\flat_{mM7}$	$B_{mM7}$

図2 コード進行一覧

4.2. コード進行の正規化

ある Block(定義2)同士の Chord を比較した際に、Chord は違うが、Chord の遷移は同じである場合がある。このような場合にも、類似したコード進行パターンとみなして抽出も行うために、コード進行の正規化を行う。楽曲の調によって、Chord のトーンの値を変える。正規化の手法は以下の通りである。

- ・「ハ長調・イ短調」の場合：トーンのまま
- ・「ト長調・ホ短調」の場合：トーンの半音7個下
- ・「ニ長調・ロ短調」の場合：トーンの半音2個下
- ・「イ長調・嬰へ短調」の場合：トーンの半音9個下
- ・「ホ長調・嬰ハ短調」の場合：トーンの半音4個下
- ・「ロ長調・嬰ト短調」の場合：トーンの半音11個下
- ・「嬰へ(変ト)長調・嬰ニ(変ホ)短調」の場合  
 : トーンの半音6個下
- ・「へ長調・ニ短調」の場合：トーンの半音5個下
- ・「変ロ長調・ト短調」の場合：トーンの半音10個下
- ・「変ホ長調・ハ短調」の場合：トーンの半音3個下
- ・「変イ長調・へ短調」の場合：トーンの半音8個下
- ・「変ニ長調・変ロ短調」の場合  
 : トーンの半音1個下

図3にへ長調の楽曲の例を示す[4]。この楽曲は、へ長調であるので、正規化を行うとトーンは半音5個下

のトーンとなり、コード進行は、「 $C, G, Am, Em, F, C, Dm7, G7$ 」となり、「(0, 0, 0), (7, 0, 2), (9, 1, 4), (4, 1, 6), (5, 0, 8), (0, 0, 10), (2, 4, 12), (7, 2, 14)」と数値化される。



図3 へ長調の楽曲

4.3. 楽曲のブロック分け

音楽では一般に4小節、8小節といった2の累乗の小節数が見つかる。本システムでは楽曲を4小節ごとに分割し、分割した4小節をブロックと名づける。小節を分割した際に、曲の最後や1番から2番に切り替わるときなどに小節が余る場合がある。その場合、余った小節や区切りの悪い小節は切り捨てて考える。そこで、ブロックおよび楽曲を以下のように定義する。

【定義2】ブロック Block

Block は  $n$  個のコード進行  $Chord_i (0 \leq i < n)$  によって以下のように構成される。

$Block = (Chord_0, Chord_1, \dots, Chord_n)$

例)  $Block_0 = \{(1, 0, 0), (3, 0, 4), (8, 0, 8), (5, 1, 12)\}$

【定義3】楽曲 M

楽曲  $M$  は  $n$  個の Block によって以下のように構成される。

$M = (Block_0, \dots, Block_n)$

4.4. 類似度

4.4.1. 近親調

コード進行の類似度に関して、我々は音楽理論にて定義されている近親調(Relative Key)をもとに定義することとした。図4に示す五度圏[5]を参考に、近親調について説明する。近親調は以下6種類に分けることができる。

(1) 平行調

同一の調号をもつ長調と短調のことである。もともになる調が長調ならば、短3度下の短調、もともになる調が短調のならば、短3度上の長調が平行調の関係である。例えば、CとAmが平行調の関係である(図4①)。

(2) 属調

もともになる調の完全5度上の調のことである。例えば、Cの属調はGである(図4②)。

(3) 下屬調

もともになる調の完全4度上の調のことである。例えば、Cの下屬調はFである(図4③)。

(4) 属調平行調

属調の平行調のことである。もともになる調が長調

ならば、長3度上の短調、もともになる調が短調ならば、長2度下の長調が属調平行調の関係である。例えば、Cの属調平行調は、E<sub>m</sub>である(図4④)。

(5) 下屬調平行調

下屬調の平行調のことである。もともになる調が長調ならば、長2度上の短調、もともになる調が短調ならば、長3度下の長調が下屬調平行調の関係である。例えば、Cの下屬調平行調は、D<sub>m</sub>である(図4⑤)。

(6) 同主調

同一の主音を持つ長調と短調のことである。例えば、CとC<sub>m</sub>が同主調の関係である(図4⑥)。

以上の近親調のうち、最も類似している関係は平行調であり、互いをもう一方の代理コードとして利用することもある。

一方最も類似度が低いものは同主調であり、同主音の長調と短調という対称的な関係であるため、他の近親調と比較しても圧倒的に「遠い」関係にある。

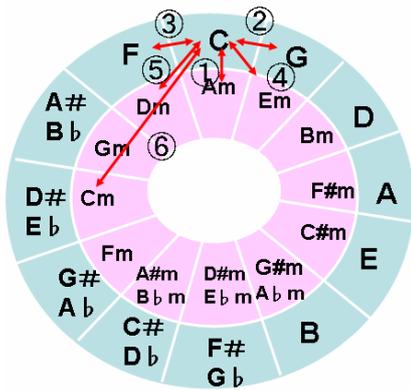


図4 五度圏

4.4.2. コードの類似度

コードは図2に示したように14種類あるが、「aug, 6, add9, sus4, (dim), (m7<sup>-5</sup>)」は「M」, 「m6, mM7」は「m」, 「7sus4」は「7」とほぼ同じであると考えられる。「sus4」と「7sus4」に関して、これらはその次のコードが同じ主音のマイナーである場合に限り、「sus4」は「m」, 「7sus4」は「m7」と解釈されるが、このような用法は頻出しないので、上記のようにまとめる。また、「dim」, 「m7<sup>-5</sup>」に関しては、「M」にも「m」にもまとめにくい例外的なものなので、現時点では「M」にまとめる。以上のことより、「M」, 「7」, 「M7」, 「m」, 「m7」の5つのコードについての類似度を考える。

図5にコードの類似度の関係図を示す。コードは長調に使うコード同士、短調に使うコード同士は類似していると考えられる。よって、「M」, 「7」, 「M7」同士は類似度が高く、また、「m」, 「m7」同士も類似度が高

い関係にある。一方、長調に使うコードと短調に使うコードの類似度は低いため「遠い」関係にある。

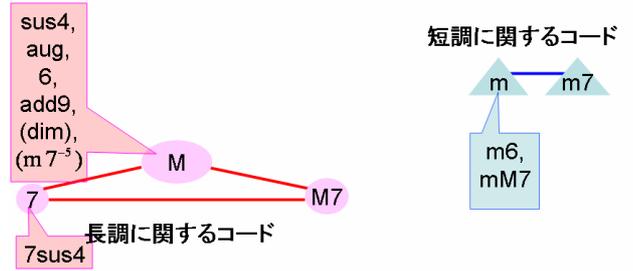


図5 コードの類似度

4.5. コード進行間の距離の定義

前節の近親調およびコード進行のコード間の関係をもとに、我々はコード進行間の距離に編集距離を利用することとした。編集距離とは、オブジェクトAにいくつかの基本編集を適用し、オブジェクトBにした時に、その編集コストを距離とすることである。我々はコード進行の基本編集を定義するために「基本コード進行変換」を導入する。

【定義4】基本コード進行変換・基本変換コスト

コード進行Aをもう一方のコード進行Bに変換することを基本コード進行変換と呼び、

$$T(\text{type}, A \Rightarrow B)$$

と表す。typeは基本変換の種類であり、調変換(定義6)、コード変換(定義8)がある。

例)CはT(平行調, C⇒A<sub>m</sub>)でA<sub>m</sub>に変換される。

また、基本変換のコストは、

$$\text{Cost}(T(\text{type}, A \Rightarrow B))$$

と表す。コストは変換の種類によって決定される。

【定義5】コード進行間の距離

コード進行a<sub>0</sub>に対して、n回の基本変換T<sub>i</sub>(type<sub>i</sub>, a<sub>i</sub>⇒a<sub>i+1</sub>)(0≤i<n-1)を施した結果コード進行a<sub>n</sub>に変換される時、コード進行a<sub>0</sub>, a<sub>n</sub>間の距離d(a<sub>0</sub>, a<sub>n</sub>)は以下の式で表される。

$$\sum_{i=0}^{n-1} \text{Cost}(T(\text{type}, a_i \Rightarrow a_{i+1}))$$

4.6. 基本編集

基本編集は、調変換とコード変換とで構成する。調変換は第4.4.1項をもとに定義する。まず、(3)下屬調は(2)属調の距離と同じものと考えられ、(4)属調平行調は属調と平行調、(5)下屬調平行調は下屬調と平行調を組み合わせたものである。また、同主調変換は平行調変換と属調変換で表すことができる。例えば、T(同主調, C⇒C<sub>m</sub>)は、(T<sub>0</sub>(平行調, C⇒A<sub>m</sub>), T<sub>1</sub>(属調, A<sub>m</sub>⇒C<sub>m</sub>))となる。なお、平行調変換を属調変換と同主調変換で表すこともできるが、この場合、最も類似した平行

調を最も「遠い」関係である同主調と属調で表すため、コスト加算による距離が成立しない。そのため、平行調を基本編集とする。

【定義 6】 調変換

(6.1) 平行調変換

あるコード進行 A が「M」の場合、短 3 度下の「m」に変換し、「m」の場合、短 3 度上の「M」変換する。

(6.2) 属調変換

あるコード進行 A の完全 5 度上のトーンに変換する。

例) A<sup>b</sup> は (T<sub>0</sub>(平行調, A<sup>b</sup> ⇒ F<sub>m</sub>), T<sub>1</sub>(属調, F<sub>m</sub> ⇒ C<sub>m</sub>)) で C<sub>m</sub> に変換される。

【定義 7】 属調の距離

コード進行 A と B の属調の距離は以下の式で定義する。

$$\sum_{i=0}^a \frac{d}{\alpha^i}$$

ただし、 $\alpha$ ,  $d$  は任意の数、 $a$  は以下の式を満たす値である。

$$B = (A + 7a) \text{ mod } 12 \quad (1 \leq a)$$

【定義 8】 コード変換

あるコード進行のコードが「aug」, 「6」, 「add9」, 「sus4」, 「dim」, 「m7<sup>-9</sup>」ならば「M」に, 「7sus4」ならば「7」に, 「m6」, 「mM7」ならば「m」に変換する。

5. 距離の正当性に関する予備実験

「ヒット曲とコード進行の関係」や「年代とコード進行の関係」といった他属性との相関関係のマイニングを行うにあたり、第 4 章で定義した距離の正当性を測るために、第 3 章、第 4 章で述べた手順・定義に従って楽曲クラスタリングシステムを構築し、J-POP の楽曲を対象に予備実験を行った。今回の予備実験では、3 人の作曲者の楽曲を 2 曲ずつ用意し、合計 6 曲、Block 数 75 個を対象にクラスタリングを行い、距離の正当性について考察した。扱ったデータに関しては、第 4.1 節で述べた定義に従って、楽譜を見ながら著者が手動で作成した。

5.1. クラスタリング手法

今回の予備実験では代表的な非階層的クラスタリング手法である k-means 法を用いた。以下にアルゴリズムを述べる。

- (1) K 個のクラスタの中心をランダムに設定する
- (2) それぞれの Block を最も近い中心に割り当てる
- (3) 全てのクラスタごとに中心を再定義
- (4) 全てのクラスタの中心が変化しなくなるまで、(2)に戻り繰り返す

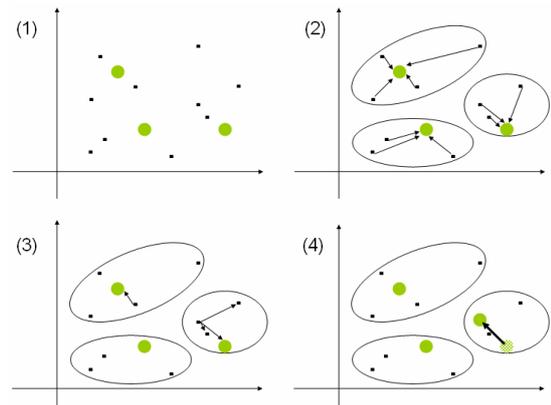


図 6 k-means 法(k=3 の場合)

図 6 に k=3 の場合についての図を示す。

本システムにおいては、それぞれの Block 間の相対的な距離は分かるが、距離を求める際に編集距離を用いているため、平均値などは求められない。クラスタの中心を再定義する際に centroid は利用できないため、クラスタの中心を再定義する際に medoid を用いた。medoid とは、クラスタ内の最大距離が最小である点のことである。

5.2. 適切なクラスタ数についての検証

今回の予備実験ではクラスタの数を変化させ数回の実験を行った。一般に一つの楽曲には、骨組みとなるコード進行が 3 種類あると考えられている。よって、本論文では、(楽曲数)×3 のクラスタが必要であると考えた。

まず、k=18 で実験を行った。この場合、性質の良いクラスタ、つまり類似したコード進行で構成されたクラスタと、雑多なクラスタ、つまり類似度は低いと同じクラスタに含まれているクラスタが混在していた。これは、k-means 法は指定された数のクラスタで分類を行わなければならない、k=18 では十分な数のクラスタ数ではないため、距離がさほど近くない場合でもいずれかのクラスタに属さなければならないということが原因であると考えられる。よって、クラスタの数を増やせば雑多なクラスタは減るのではないかと考え、(楽曲数)×3n でクラスタの数を増やして実験を行った。ただし Block 数以上のクラスタにならない範囲で実験を行った。k=36 の場合、k=18 では雑多であったクラスタの内の Block が分散し、新たに形成されたクラスタに属しているものが見受けられ、性質の良いクラスタが多く存在していた。k=54 の場合も同様に、k=36 の場合に比べ雑多なクラスタが分散された。しかし、性質の良いクラスタもさらに分割され、より細かいクラスタが出来上がった。k=72 の場合は、クラスタ数が Block 数とほぼ変わらないため、一つのクラスタ内の要素数が少なく、全体的に分散した結果となった。ま

clusterNo.	musicNo.	blockNo.	コード進行	★1との距離	★2との距離	★3との距離
	11	4	11 Am Am Am Am Em Em Em Em FFGGCCCC	42.0	601.4	476.0
★1	11	2	9 Am Am Am Am Am7 Am7 Am7 Am7 FFGGCCCC	-	531.4	412.2
	11	2	7 Am Am Am Am Am7 Am7 Am7 Am7 FFGGCCCC	0.0	531.4	412.2
	11	0	12 Am Am AmM7 AmM7 Am7 Am7 Am6 Am6 FFFF Dm7 Dm7 Dm7 Dm7	191.4	491.6	521.6
★2	28	4	5 FFFF CCCC Dm7 Dm7 Dm7 Dm7 Dm7 Dm7 GG	531.4	-	151.0
	28	0	1 FFFF CCCC Dm7 Dm7 Dm7 Dm7 Dm7 Dm7 Ab Bb	545.2	36.9	187.7
	28	4	6 FFFF CCCC Dm7 Dm7 Dm7 Dm7 Dm7 Dm7 G7 G7	533.4	2.0	153.0
	30	0	5 FFFF CM7 CM7 CM7 CM7 Dm7 Dm7 Dm7 Dm7 CCCC	431.6	104.2	42.2
	30	2	5 CCCC Am Am Am Am Dm Dm Dm Dm GGGG	313.2	253.2	42.4
★3	30	3	6 FFFF C C Am Am Dm7 Dm7 Dm7 Dm7 GGGG	412.2	151.0	-

図 7 性質の良いクラスタの例

clusterNo.	musicNo.	blockNo.	コード進行	★4との距離	★5との距離
	1	2	4 CCCC Bm7-5 Bm7-5 E7 E7 Am Am Am Am7 DDDD	128.3	195.0
★4	1	0	8 FFFF C C E7 E7 Am Am Am D7 D7 D7 D7	-	323.2
	6	5	7 FFFF Em Em Am Am Dm Dm C C FFGGCCCC	313.0	382.7
	6	4	0 Am Am Am Am Em7 Em7 Em7 Em7 FFGGCCCC	267.0	145.0
★5	6	4	3 FFFF Em7 Em7 Em7 Em7 FFFF Gsus4 Gsus4 GG	175.3	-
	6	0	9 FFC C Dm7 Dm7 Dm7 Dm7 C Bb Bb Ab FFGG	399.1	415.4

図 8 雑多なクラスタの例

た  $k=54$  の場合と同様に、似ているコード進行でも別のクラスタに属しているものもあり、必要以上に分割されるという結果になった。

以上のことより、今回の実験では雑多なクラスタが少なく、性質の良いクラスタも多く存在していた  $k=36$  でクラスタリングを行った場合に性質の良い結果が得られるということが分かった。

### 5.3. 距離の正当性

図 7 に、 $k=36$  でクラスタリングを行った際の特に性質の良い結果が得られたクラスタの例、図 8 に雑多なクラスタの例を示す。図中の「★」は各クラスタの中心の Block であることを示す。

性質の良いクラスタにおいて、同じクラスタに含まれるコード進行を比較してみると、複数並んでいるコードのうち数個のコードだけが違うという場合と、Block 内の要素同士は別のものであるが、第 4 章にて定義した距離の近いもの同士が集まっているという結果になった。後者の場合について、例えば、図 7 の「clusterNo.30, musicNo.2, blockNo.5」のコード進行に関して、同じクラスタ内の「clusterNo.30, musicNo.3, blockNo.6」のコード進行を比較すると、「F と C」は属調関係、「C と Am」は平行調関係にあり、また、「Dm7 と Dm」はどちらも短調のコードであり、距離として「近い」関係にある。つまり、この二つの Block に関しては距離が「近い」、つまり類似したコード進行であるといえる。また、このことについて、実際にコードを演奏し聴いてみたところ、同じクラスタに含まれるブロック同士のコードは実際に人間が耳にした場

合も似ていると認識するコードであることが確認できた。また、性質の良いクラスタと雑多なクラスタについて、各クラスタの中心とそのクラスタ内の他の Block の距離の大きさについて比較したところ、性質の良いクラスタは、中心からの距離が平均 46.5、雑多なクラスタは、中心からの距離が平均 221.3 となった。

また、clusterNo.11 の blockNo.9 および 7 のコード進行は「Am・F・G・C」で構成されており、藤井[4]によると、このコード進行は、「T(トニック)-S(サブドミナント)-D(ドミナント)-T(トニック)」という進行になっていると言える。よって、音楽理論で代表的なコード進行パターンと言われるコード進行で構成されたクラスタが出来ることが確認できた。

以上のことにより、我々が定義した、近親調を用いたコード進行の距離は正当性が高いと考えられる。

### 5.4. コード進行類似性の意味

今回の予備実験により、同一の楽曲のコード進行は同一のクラスタに多く含まれるという傾向が見られた。このことにより、同じ楽曲中には類似したコード進行が多く用いられるのではないかと考えられる。今回の実験は、距離の正当性に関しての予備実験であったため、データ数を絞って実験を行ったので上記の考察は参考程度であるが、今後、データ数を増やし、上で述べたことの正当性の更なる検証を行っていく。

## 6. まとめと今後の課題

本論文では、コード進行の数値化、コード進行の正規化、楽曲の分割、近親調を用いたコード進行の類似

度の定義，距離の定義を行い，k-means 法を用いたクラスタリングシステムを実装し，距離の正当性に関する予備実験を行った．予備実験により，データに対して適切な数のクラスタを用意してクラスタリングを行うと，性質の良いクラスタが多く得られることが分かった．また，クラスタリング結果により，近親調を用いたコード進行の距離の定義の正当性が高いことが確認できた．しかし，楽曲のクラスタリングを行う際，k-means 法はいくつのクラスタを用意すべきか予測が困難でありクラスタの数を換え繰り返し実験を行わなければならないかった．

今後は楽曲データ数をさらに増やし，楽曲のクラスタリングに最適なクラスタリング手法を用いて，より性質の良いクラスタリングを行う．クラスタリング手法には，階層的クラスタリングの適用を考えている．階層的クラスタリングは，クラスタの数を指定する必要はなく，距離が近いもの同士を同じクラスタに統合していくため，k-means 法よりも，より良い分類結果を得ることが期待できる．また，コード進行の数値化に関して，現時点では四分音符を1拍と定義しているが，今後1拍の定義に関して最適な長さについて検討して行く．また，人間が耳にした際に似たコードであると感じるかについて，多くの人に聞いてもらい実験を行う．さらに，クラスタリング結果をもとに，作曲家，楽曲名，発売日，その楽曲のヒットチャートといった属性を追加し，「ヒット曲とコード進行の関係」，「年代とコード進行の関係」といった他属性との相関関係の抽出を行っていく予定である．また，コード進行以外のメロディやリズムパターンといった音楽的特徴に関してもクラスタリングを行い，楽曲傾向をマイニングしていく．

## 文 献

- [1] 松田卓久，飯島淳一，“テキストマイニング技術の音楽情報への適用”，経営情報学会 2002 年春季全国研究発表大会予稿集，pp.186-189，2002，
- [2] 梶克彦，“音楽におけるアノテーションとその応用に関する研究”，名古屋大学大学院修士論文，2004
- [3] 吉原一期，近山隆，“Web 上の演奏データを利用した作曲支援システム”，情報処理学会第 65 回全国大会，pp.2-197-2-198，2003
- [4] 藤井英一，“藤井英一のコード進行講座”，株式会社ヤマハミュージックメディア，103p.，2006
- [5] 北川祐，“ポピュラー音楽理論”，リットーミュージック，278p.，2004