

シーケンシャルパターンマイニング拡張による 特徴的なコード進行の抽出手法

篠原 透^{1,a)} 沼尾 雅之^{1,b)}

概要: 音楽におけるコードは、メロディーを引き立て、楽曲全体の調和を保つ重要な要素である。しかし、作曲の初心者においては、音楽理論をもとにこれを適切に構成することは困難である。また、作曲経験者にとっても、終止形などの特定のコード進行を重要視したり、特定の作曲者の楽曲に似せるためにコード進行を模倣したりすることがある。本稿では頻出パターンマイニングの一種であるシーケンシャルパターンマイニングを逆順に走査したり、独立した閾値を用いて疑似アイテムを制御できるよう拡張したりすることにより、音楽情報処理に特化したパターンマイニング手法を提案する。本提案手法を市販の J-POP やロック、洋楽を含む楽曲 100 曲のコード進行データに適用したところ、作曲者の特徴を表すコード進行や終止形のパターンを抽出することができたので、その結果を報告する。

キーワード: シーケンシャルパターンマイニング, 音楽情報検索, 作曲支援

The Algorithm to Extract Characteristic Chord Progression Extended the Sequential Pattern Mining

SHINOHARA TORU^{1,a)} NUMAO MASAYUKI^{1,b)}

Abstract: Chord is an important element of music because it supports melody and harmony of whole music. However, it is difficult to compose the chord progression using the music theory efficiently for the beginner of music composer. In addition, it is conceivable that professional composers often regard specific chord progression as important, such as cadenza pattern, and they imitate the chord progression used by the other composers. In this paper, we suggest the pattern mining method specified for music information processing by extending the sequential pattern mining, one of the frequent pattern mining so that it can scan the data in opposite order and it can process pseudo items using independent threshold. We report that chord progressions representing character of composers and cadenza patterns are extracted after we apply the method from 100 musics including J-pop, rock and European music.

Keywords: Sequential Pattern Mining, Music Information Retrieval, Computer-aided Composition

1. はじめに

近年, DeskTop Music (DTM) の普及によって楽器演奏技術を持たないユーザーでも作曲に触れることができるようになったが, 実際にどのような楽曲を制作するかは作曲

者自身の経験と技術次第であり, 一定の音楽的な知識を必要とする。その中でも, コードに関する音楽理論は難解であり, 作曲を始めたばかりの初心者が使いこなすことは困難である。また, 作曲経験者の場合は, 終止形などの特定のコード進行を重要視して作曲を行ったり, 特定の作曲者の楽曲に似せる目的でコード進行を模倣するなどの需要が考えられる。したがって, ポピュラー音楽に使われる特徴的なコード進行をデータベースから抽出し, 多彩なコード進行を直感的かつ適切に構成できることは, 作曲において

¹ 電気通信大学大学院 情報理工学研究所
Graduate School of Informatics and Engineering, The University of Electro-Communication

a) toru.shinohara@uec.ac.jp

b) numao@cs.uec.ac.jp

重要な課題であると考えられる。

実際の楽曲から繰り返し部分を抽出したり、特徴的な部分を抽出するマイニング方法には、2種類のアプローチが考えられる [1]。一つは音楽音響信号を直接解析対象としてマイニングする手法である。もう一つは、音楽的情報をテキストなどの記号で表現したものに對し、テキストマイニングの手法を適用するものである。本研究では後者により、多彩なコード進行を抽出することを考える。

音楽的情報を記号処理的にマイニングした研究例として、藤川らは音楽音響信号に対して短時間のクロマベクトルを算出し、得られた音高を記号に変換したものに對するマイニングを行い、楽曲の繰り返し部分を抽出する手法を提案した [2]。また、Jiménez らは MusicXML からメロディーラインを記号列化したものをマイニングし、楽曲の主題 (motifs) を抽出した [3]。

コード進行に特化してマイニングを試みた研究例として、Ogihara らは楽曲のコード進行についての N-gram 頻度から作曲者を表す類似度を定義し、これをもとにクラスタリングを行った [4]。また、Pérez らも楽曲のコード進行についての N-gram 頻度から楽曲のジャンルを推定する手法を提案した [5]。N-gram 頻度はテキストマイニングのみならず、コード進行の分析に有効な手法の一つである [6]。しかし、対象とするデータは専らコードのみであり、付加的な要素を考慮してコード進行を提示することを目的としたものではない。

著者らはかつて、実際の楽曲のコード進行から抽出される特徴的なコード進行を作曲支援に有効に活用する手段として、ナイーブなシーケンシャルパターンマイニングを用いてコード進行を提示する作曲支援システムを提案した [7]。この研究では、作曲を始めたばかりの初心者を対象とした被験者実験により、提案したシステムを用いて制作時間の短縮を図ることができることを示したものの、抽出されたコード進行の中で特に有用と考えられるものは、コードの進行の仕方やコードの複雑度などに依存していることが示された。

本稿では、ナイーブなシーケンシャルパターンマイニングを拡張し、コードの進行の仕方に特化した探索を行った。非コード属性をもったコード進行の情報を扱えるようにすることにより、実際の楽曲に含まれる特徴的なコード進行を発見し顕在化できることを示す。

2. 既存マイニング手法

2.1 シーケンシャルパターンマイニング

シーケンシャルパターンマイニング [8] とは、1995 年に Agrawal らによって提唱された問題で、時系列に配置されたデータ列から頻出パターン系列を抽出するためのデータマイニング手法のことである。

出現する要素の集合を $\Sigma = \{i_1, i_2, \dots, i_m\}$ と表し、 i_j を

アイテムと呼ぶ。このとき、時系列のデータ列 S を**シーケンス**と呼び、 $S = \{ \langle a_1, a_2, \dots, a_r \rangle \mid a_i \in \Sigma \}$ と表す。また、シーケンスのデータベースを**トランザクション**と呼び、 $T = \{S_1, S_2, \dots, S_n\}$ と表す。シーケンス α 中の全てのアイテムが、別のシーケンス β 中に存在し、さらにその順序が保持されている場合、シーケンス α をシーケンス β の部分シーケンスであると定義し、 $\alpha \subseteq \beta$ と表す。ここで、トランザクション T を構成する全てのシーケンスのうち、部分シーケンス α の出現回数を α の support 値と定義する。

$$support(\alpha) = |\{\alpha \mid \alpha \subset S\}| \quad (1)$$

また、シーケンス α がシーケンス β の部分シーケンスで、 $\alpha = \langle a_1, \dots, a_l \rangle, \beta = \langle a_1, \dots, a_l, a \rangle$ となるとき、 α の出現回数に対する β の出現回数の比率を、 α から β への confidence 値と定義する。

$$confidence(\alpha \rightarrow \beta) = \frac{support(\beta)}{support(\alpha)} \quad (2)$$

シーケンシャルパターンマイニングとは、任意の閾値 (**最小 support 値, minimum support**) ζ に対し、 $support(\alpha) \geq \zeta$ となるような部分シーケンス α を全て列挙する問題のことをいう。このような α を**頻出パターン**と呼ぶ。

抽出された頻出パターンは時系列的な順序を保っているものの、必ずしも隣接したアイテムの関係を抽出するわけではない。頻出パターンにおける不連続アイテム間のことを**ギャップ**と呼ぶ。

2.2 アルゴリズム

シーケンシャルパターンマイニングを効率よく行うアルゴリズムとして、2001 年に Pei らは PrefixSpan [9] を提案した。この手法では、prefix projection という射影方法とその射影によって生成される射影データベースを用いて、頻出パターンを深さ優先的に探索する。

射影元のシーケンス $S = \langle a_1, a_2, \dots, a_l \rangle$ と射影対象のアイテム a に対し、 $a_1 \neq a, a_2 \neq a, \dots, a_{j-1} \neq a, a_j = a$ となるような整数 $j (1 < j < l)$ が存在するとき、部分シーケンス $\langle a_1, \dots, a_j \rangle$ を S の a に対する prefix ($prefix(S, a)$)、部分シーケンス $\langle a_{j+1}, \dots, a_m \rangle$ を S の a に対する postfix ($postfix(S, a)$) と定義する。あるトランザクション T の各シーケンスに対してアイテム a を射影し、それらを改めてパターンのデータベースとしたものを**射影データベース** (projected database) $T|a$ と定義する。

$$T|a = \{S' \mid (S \in T) \wedge (S' = postfix(S, a)) \wedge (S' \neq \epsilon)\} \quad (3)$$

PrefixSpan の大まかなアルゴリズムは以下のように示される。

(1) トランザクション T の走査によって長さ 1 のパターンを数え上げ、最小 support 値を満たすもののみを抽出

する。

- (2) 生成された射影データベースに対して再帰的に走査を行い、最小 support 値を満たすものを長さ 2 のパターンとして抽出する。
- (3) 以降、同様にして射影データベースを走査して長さの長いパターンを抽出し、最小 support 値を満たさなくなるまで繰り返す。

このとき、最小 support 値を満たさなかったパターンを棄却し、再帰的探索の対象から除くことを**枝刈り**する、と呼ぶ。

以上は原著の PrefixSpan アルゴリズムであるが、本稿ではさらにパターンの長さやギャップの長さ、ギャップの出現回数などを細かく指定し、枝刈りの条件として拡張したものの [10][11] をナイーブなシーケンシャルパターンマイニングとして扱う。

3. 提案手法

3.1 コード進行抽出問題への適用

本稿で取り扱うマイニングにおいて、アイテム i_j がコード c_j に、シーケンス S が一つの楽曲 M に、トランザクション T が楽曲データベース D に対応する。また、 D において、あるコード進行 P が m 回出現するとき、 m は P の support 値 $support(P)$ である。ここでは、 P が閾値として設定した ζ 回以上出現した場合に頻出なコード進行として抽出する問題として定義する。一つの楽曲 M 上に同一のコード進行 P が t 回登場する場合は、 t 回として数えられる。

ここで、アイテムとなるコードは C, Am などの絶対表記ではなく、I から VII のローマ数字による和音記号を用いた相対表記を用いる。入力を絶対表記で受け取った場合は、その楽曲の調を基準として変換を行う。なお、曲の途中で転調がある場合は、その時点から転調先の調に基づいて変換を行うものとする。

support 値が高いものは一般的に使われるコード進行、support 値が低いものはユニークなコード進行を意味する。確率的な最適化手法と異なり、コード進行が一般的に使用されているという妥当性を保ちながら、音楽的に珍しいコード進行や意図的な音楽的表現を損なわずに抽出することができる。

ただし、ナイーブなシーケンシャルパターンマイニングでは、一般的に使われる support 値の高いコード進行のみが多く抽出され、ユニークなコード進行が抽出されないケースが目立つ。そこで、提案システムでは以下に述べる工夫を施すことにより、特定のコード進行の抽出を効率よく行うことや、作曲家ごとの頻出なコード進行のパターンの抽出を可能にする。

3.2 終止形パターン発見のための逆順走査手法

通常、シーケンシャルパターンマイニングは楽曲 $P = \langle c_1, \dots, c_m \rangle$ に対し、 c_1 から c_m に向かって走査を開始する。これを c_m から c_1 に向かって逆順に走査することを考える。

このとき、アイテム $c_j = I$ に対して頻出パターン系列を探索することにより、終止形のコード進行に関する頻出なパターンを抽出することができる。このパターンには完全終止 $\langle V, I \rangle$ や変終止 $\langle IV, I \rangle$ などが含まれている。

この場合、confidence 値は頻出パターン $\alpha = \langle a_1, \dots, a_1 \rangle, \beta = \langle a, a_1, \dots, a_1 \rangle$ に対して、 α の出現回数に対する β の出現回数の比率であり、正順走査したときの値とは異なるため、以下正順走査時の confidence 値を正順 confidence 値、逆順走査時の confidence 値を逆順 confidence 値と呼称する。正順と逆順を記述しない場合は、正順 confidence 値を表すものとする。

3.3 疑似アイテム付加による非コード属性の取り扱い

ここまでの定義では、アイテムがコードのみである場合を示したが、これに疑似アイテムを付加し、1つの楽曲についている属性を含めてマイニングが行えるよう拡張する。ここでの疑似アイテムは楽曲の作曲家、ジャンル、制作年、制作地などを想定しており、これによって属性を考慮した頻出パターンが取り出せると考えられる。ここで、楽曲中のコードを表すアイテムを疑似アイテムと区別して**コードアイテム**と呼称する。

楽曲 M に付加する N 個の疑似アイテムの集合を $\Sigma'_M = \{pseudo_1, \dots, pseudo_N\}$ とするとき、楽曲データベース D 中の楽曲 $M = \langle c_1, \dots, c_m \rangle$ に対し、先頭に疑似アイテムを付加する。具体的には、正順走査の場合は、 $M' = \langle pseudo_1, \dots, pseudo_N, c_1, \dots, c_m \rangle$ 、逆順走査の場合は $M' = \langle c_1, \dots, c_m, pseudo_N, \dots, pseudo_1 \rangle$ とした楽曲データベース D' を作る。その上で、疑似アイテムとコードアイテムをアイテム i_j と同一視して数え上げることを考える。

この場合、作曲家アイテムは走査時に始めに探索されることになる。提案手法では、疑似アイテム $pseudo_k$ に対する射影データベース $M|pseudo_k$ を走査し、アイテムがコードアイテムである場合にのみ、ギャップを許容する。なお、この場合に限り、隣接するコードアイテム (c_1 または c_m) を数え上げる際も、長さ 0 のギャップが含まれているものとし、隣接しているかどうかを考慮しないものとする。

疑似アイテム $pseudo_k$ の出現回数は 1 楽曲につき高々 1 回であるが、コードアイテム c_j はその限りではない。したがって、共通の最小 support 値を設定して抽出を行った場合、相対的に低い support 値をもつ $pseudo_k$ を含むパターンは抽出されにくくなる。

そこで、最小 support 値 ζ とは別に、各作曲家ごとの

最小 support 値 ζ_{pseudo_k} を定義する。ここでは一般的に ζ_{pseudo_k} は ζ よりも低く設定し、走査の段階で疑似アイテム $pseudo_k$ を発見した場合は、 ζ ではなく ζ_{pseudo_k} を優先して適用する。これにより、 $pseudo_k$ を含むパターンが枝刈りされることを防ぐことができる。

なお、本稿では作曲者ごとのコード進行の違いを抽出するために、作曲者を表すアイテムを導入する。これを**作曲者アイテム**と呼称し、 $comp_k$ と表す。

3.4 コード進行の繰り返しパターンの抑制

例えば、出力パターンとして以下のようなコードが与えられたとする。

$$\langle V, I, IV, I, IV, I, IV, I, IV \rangle \quad (4)$$

楽曲によっては、このようにコード進行が繰り返されるケースがある。これらは音楽的知識の直感的な発見という観点からみると、変化に乏しく他の頻出パターン系列によって補完することができるパターンであるといえる。また、むやみに出力パターンの数が増大することは、計算量の増大にも繋がる。そこで、任意のコード進行 $p \neq \epsilon, q \neq \epsilon$ に対して、次のようなコード進行 P を枝刈りの対象とし、出力を抑制する。なお、 $*$ は 0 回以上の繰り返し、 $+$ は 1 回以上の繰り返しを表す。

$$P = \langle q * pp+ \rangle \quad (5)$$

このようなパターンを、本稿では**繰り返しパターン**と呼称する。先の例 $P = \langle V, I, IV, I, IV, I, IV, I, IV \rangle$ の場合、 $p = \langle I, IV \rangle, q = \langle V \rangle$ となる。このとき、枝刈りによって $\langle V, I, IV, I \rangle$ までが頻出なコード進行として出力される。

また、この定義にならうことにより、任意のアイテム a が連続する、 $\langle a, a \rangle$ のようなパターンも枝刈りの対象となる。

3.5 条件のまとめ

以上の条件やその他シーケンスの出力に関する条件をまとめると、次のようなパターンをコード進行として抽出することとなる。

- support 値が ζ 以上で、疑似アイテムの support 値が ζ_{pseudo_k} を満たす。
- パターンの長さは 2 以上である。
- ギャップを含まない隣接したパターンのみを抽出する。ただし、疑似アイテムが先頭にあるパターンの場合は、その直後のコードアイテムに対してのみ長さ 0 以上のギャップを許容する。
- 同じパターンの繰り返しとなる冗長なパターンの出力を抑制する。

4. 評価実験

4.1 作曲者を含むパターンの抽出実験

4.1.1 目的・方法

作曲者アイテムを含むコード進行が、作曲者の特徴を適切に表しているかどうかを示す。作曲者ごとに特徴がある場合、作曲者アイテムのない通常のパターンに比べ、作曲者アイテムを含むパターンは confidence 値に有為な差があると考えることができる。confidence 値は、ある頻出パターン $\langle i_1, \dots, i_m \rangle$ に対して、後続のアイテム i の分布の比率であるから、帰無仮説を「作曲者アイテムを含むコード進行と含まないコード進行では、分布の比に有意差はない」とし、有意水準を 5% とした χ^2 検定により示す。最小 support 値は $\zeta = 2, \zeta_{comp_k} = 1$ とし、可能な限りパターンを抽出した。

ここでは、正順走査における頻出パターン $P_1 = \langle I \rangle$ と $P_2 = \langle comp_k, I \rangle$ に対して、後続の頻出コードアイテム c_j の support 値を一つの分布として捉える。一般に、 I はトニックと呼ばれ、どのコードに進行しても禁則とはならないため、後続の進行の仕方によって作曲者に特徴が出るものと考えられる。

その他に、作曲者を含むパターンで confidence 値が高いものは有為差が高いかどうかを調べるために、作曲者アイテムを含むパターンのうち、support 値の高い順に confidence 値が最大値である 1.00 となるものについても χ^2 検定を行った。

実験には、現在 YAMAHA 社がメタデータを含む MIDI (XF) 形式として販売している、J-POP やロック、洋楽を含む楽曲 100 曲のコード進行をデータベースとした。また、対象とするコードの種類も XF 形式がサポートするものを対象とし、分数コード表記にも対応した。

4.1.2 結果・考察

コード進行 $\langle I \rangle$ に対する後続のコード進行に対して、表 1 の左側に検定した結果の一部を示す。なお、 $s = support(\langle comp_j \rangle)$ である。また、表 1 の作曲者において最も正順 confidence 値の高かったパターンを表 1 の右側に示す。

以上の結果から有意差はあり、帰無仮説は棄却される。したがって、作曲者アイテムを付加してマイニングを行うことにより、コード進行 $\langle I \rangle$ について抽出されたパターンは作曲者に特徴的なパターンであることが確かめられた。また、作曲者アイテムのない通常のパターンに比べて、総じて confidence 値が高いことが確認できた。

次に、confidence 値が 1.00 となる作曲者アイテムを含むコード進行に対して、表 2 の左側に実際のコード進行、右側に検定の結果の一部を示す。なお、 $s = support(\langle comp_j \rangle)$ である。

表 1 コード進行 < I > に対する後続のコード進行の有意差と confidence 値

$comp_j$	s	χ^2 値	生起確率	有意差	コード進行	正順値	逆順値
福山 雅治	3	177.62	1.68×10^{-10}	あり	< I, V/VII >	0.21154	0.61111
徳永 英明	3	178.40	1.33×10^{-10}	あり	< I, V/VII >	0.31818	1.00000
Utada Hikaru	3	262.49	6.59×10^{-23}	あり	< I, VI _m >	0.52941	0.50000
坂本 龍一	3	303.47	1.47×10^{-29}	あり	< I, III/ ^b VI >	0.39394	0.61905
アンジェラ・アキ	2	212.74	2.38×10^{-15}	あり	< I, V >	0.31081	0.47917
小室 哲哉	2	222.21	9.79×10^{-17}	あり	< I, VI _m >	0.61538	0.60377
菅野 よう子	2	467.75	1.64×10^{-58}	あり	< I, V _{7sus4} /II >	0.18519	0.55556
平井 堅	2	502.91	3.59×10^{-66}	あり	< I, IM9/III >	0.44444	0.88889
作曲家アイテムなし	-	-	-	-	< I, V/VII >	0.14491	0.71141

表 2 confidence 値が 1.00 となる作曲家アイテムを含むコード進行とその有意差

$comp_j$	P	s	χ^2 値	生起確率	有意差
福山 雅治	< I ₇ , IV _{M7} >	3	6.60	4.72×10^{-1}	なし
徳永 英明	< V ₆ , IV >	3	10.18	1.71×10^{-2}	あり
Utada Hikaru	< ^b VII _{aug} , ^b VII ₆ >	3	0.00	1.00	なし
坂本 龍一	< VI _{m7} , VI _{m7} /V >	3	96.70	3.35×10^{-6}	あり
アンジェラ・アキ	< I _{add9} /III, IV _{M9} >	2	9.91	2.71×10^{-1}	なし
小室 哲哉	< III/ ^b VI, VI _m >	2	16.36	9.55×10^{-4}	あり
菅野 よう子	< II ₇ , II _{m7} >	2	35.94	3.31×10^{-4}	あり
平井 堅	< IM9/III, IV >	2	0.00	1.00	なし

以上の結果から、有為差が認められるものと、認められないものが存在することが分かった。有為差が認められないと判定されるパターンのうち、 $\chi^2 = 0$ となるものは、作曲家を含むパターンと含まないパターンの分布が全く同じであることを意味する。これは、当該コード進行がデータベース D 中にもその作曲者の楽曲にしか存在せず、作曲家アイテムを含まないパターンと差別化できないケースである。このようなケースは、データベース D の楽曲数の増加に伴って進行の仕方が多岐にわたった場合、適切に作曲家の特徴を示すパターンになると考えられる。逆に、楽曲数を増加させても進行の仕方があまり変わらないものについては、枝刈りを行うなどの工夫も必要になると考えられる。

今回は $s = 2$ または $s = 3$ の場合の一例のみを示したが、データベース中の同じ作曲者の楽曲が少なすぎている可能性も考えられるため、今後データベースの量を増加させていく場合には慎重に検討する必要があるといえる。

4.2 終止形を含むパターンの抽出実験

4.2.1 目的・方法

4.1 節と同様にして、終止形を含むパターンについても、作曲家アイテムを含むコード進行が、作曲家の特徴を適切に表しているかどうかを示す。

ここでは、逆順走査における頻出パターン $P_1 = < I >$ と $P_2 = < I, comp_k >$ に対して、それより前にある頻出コードアイテム c_j の support 値を一つの分布として捉える。4.1 節とは異なり、ドミナント進行 < V, I > など、音楽理論上広く使用される終止形となるパターンが含まれている。

4.2.2 結果・考察

表 3 の左側に検定した結果の一部を示す。なお、 $s = support(< comp_j >)$ である。また、これらの作曲家において最も逆順 confidence 値の高かったパターンを表 3 の右側に示す。

以上から、一部有為差がないと判断されるコード進行が抽出されたことがわかった。すなわち、当該作曲家の分布は作曲家アイテムを含まない一般的なパターンの分布に近いということである。実際に有意差がないと評価された作曲家アイテムを含むパターンにおいては、confidence 値の最も高いパターンはコード進行 < V, I > で一致していることから、主な要因はこの進行が多用されていることにあると考えられる。これは、当該作曲家が一般的に使用されるコード進行を多用するという特徴を表しているともできる。一方、有為差があるにも関わらず confidence 値の最も高いパターンが作曲家アイテムを含まない場合と同じであるものは、一般的な終止形を使用するが、他にも多くの終止形を使用しているということがいえる。

5. おわりに

本稿では、シーケンシャルパターンマイニングの拡張によって、特徴あるコード進行を抽出することができることを報告した。また、シーケンシャルパターンマイニングを楽曲のコード進行データの逆から走査を行うことにより終止形を導き出せること、作曲家を表す疑似アイテムを楽曲のコード進行に付加し、独立した閾値を用いて制御することにより、作曲家ごとの特徴を表すコード進行を抽出可能であることを確認した。

表 3 コード進行 < I > に終止するコード進行の有意差と confidence 値

$comp_j$	s	χ^2 値	生起確率	有意差	コード進行	正順値	逆順値
福山 雅治	3	115.39	6.80×10^{-7}	あり	< V, I >	0.50000	0.28846
徳永 英明	3	218.85	1.13×10^{-22}	あり	< IV, I >	0.48148	0.29545
Utada Hikaru	3	429.03	3.32×10^{-61}	あり	< Π_{m7}, I >	1.00000	0.52941
坂本 龍一	3	62.76	1.25×10^{-1}	なし	< V, I >	0.50000	0.63636
アンジェラ・アキ	2	150.36	9.83×10^{-12}	あり	< IV, I >	0.84375	0.36486
小室 哲哉	2	50.53	4.92×10^{-1}	なし	< V, I >	0.60938	0.75000
菅野 よう子	2	346.42	1.57×10^{-45}	あり	< V_{7sus4}, I >	0.80769	0.38889
平井 堅	2	231.29	8.61×10^{-25}	あり	< $\Pi_{m7}/V, I$ >	0.50000	0.44444
作曲家アイテムなし	-	-	-	-	< V, I >	0.36371	0.31100

今回は作曲者に関する情報を疑似アイテムとして付加したが、今後の課題として、さらに小節の区切りや時間情報、その他楽譜情報をデータベース中のアイテムとして取り入れられるよう改良していく。また、コードの役割や複雑度に応じてアイテムそのものを階層化するマイニングの導入を検討している。現状では、コードの I, I_{M7} , I_{M9} などは全て別のアイテムとして扱われているが、音楽理論上の役割からすれば、これらは似通って使用されるコードであり、I と同族であるとみなして数え上げを行うなどの工夫が妥当であるものと考えられる。

以上の工夫を加え、抽出できるパターンに幅を持たせたり、新たに制約として加えたりすることで、より興味深い特徴の抽出・分析を進めていく。

参考文献

- [1] Cambouropoulos, E., Crawford, T. and Iliopoulos, C.: Pattern processing in melodic sequences: Challenges, caveats and prospects, *Computers and the Humanities*, Vol. 35, No. 1, pp. 9–21 (2001).
- [2] 藤川純平, 喜田拓也: 頻出直列エピソードマイニング手法を用いた音楽データからの繰り返し部分抽出, *DEIM Forum 2012 F6-4* (2012).
- [3] Jiménez, A., Molina-Solana, M., Berzal, F. and Fajardo, W.: Mining transposed motifs in music, *Journal of Intelligent Information Systems*, Vol. 36, No. 1, pp. 99–115 (2011).
- [4] Ogihara, M. and Li, T.: N-gram chord profiles for composer style representation, *Proceedings of the International Conference on Music Information Retrieval (ISMIR) 2008-Session 5d-MIR Methods* (2008).
- [5] Pérez-Sancho, C., Rizo, D. and Inesta, J. M.: Genre classification using chords and stochastic language models, *Connection science*, Vol. 21, No. 2-3, pp. 145–159 (2009).
- [6] Scholz, R., Vincent, E. and Bimbot, F.: Robust modeling of musical chord sequences using probabilistic N-grams, *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, IEEE, pp. 53–56 (2009).
- [7] 篠原 透, 大野将樹, 沼尾雅之: 作曲支援のためのマイニングによる和声進行提示システムの提案, 情報処理学会第 75 回全国大会, pp. 2–285–2–286 (2013).
- [8] Agrawal, R. and Srikant, R.: Mining Sequential Patterns, *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*, IEEE, pp. 3–14 (1995).
- [9] Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U. and Hsu, M.: PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth, *ICDE'2001, April*, pp. 215–24 (2001).
- [10] Kitakami, H., Kanbara, T., Mori, Y., Kuroki, S. and Yamazaki, Y.: Modified PrefixSpan method for motif discovery in sequence databases, *PRICAI 2002: Trends in Artificial Intelligence*, Springer, pp. 482–491 (2002).
- [11] Antunes, C. and Oliveira, A. L.: Generalization of pattern-growth methods for sequential pattern mining with gap constraints, *Machine Learning and Data Mining in Pattern Recognition*, Springer, pp. 239–251 (2003).