

## 確率的手法に基づく旋律分類に関する研究

吉原, 幸輝 / YOSHIHARA, Yukiteru

---

(発行年 / Year)

2007-03-24

(学位授与年月日 / Date of Granted)

2007-03-24

(学位名 / Degree Name)

修士(工学)

(学位授与機関 / Degree Grantor)

法政大学 (Hosei University)

2006 年度 修士論文

# 確率的手法に基づく旋律分類に関する 研究

STUDIES ON CLASSIFYING MELODIES BASED ON  
PROBABILISTIC METHOD

指導教官 三浦 孝夫 教授

法政大学大学院工学研究科  
電気工学専攻修士課程

吉原 幸輝

Yukiteru YOSHIHARA

# 目次

第1章	序論	5
1.1	扱う問題	5
1.2	背景と関連研究	5
1.3	必要とされる機能	6
1.3.1	単旋律情報に関する分類	6
1.3.2	複旋律情報に関する分類	7
1.3.3	旋律情報間の整合性	7
1.3.4	複旋律情報からのコード推定	7
1.4	論文の構成	8
1.5	発表論文	8
1.5.1	論文誌	8
1.5.2	口頭発表	9
第2章	EM アルゴリズムを用いた旋律分類	10
2.1	前書き	10
2.2	旋律の表現	11
2.3	旋律特徴量	12
2.3.1	ピッチ輪郭	13
2.3.2	ピッチスペクトル	14
2.3.3	本稿で提案する旋律特徴量	15
2.4	単純ベイズ法とEM アルゴリズム	17
2.4.1	単純ベイズ法による文書分類	17
2.4.2	EM アルゴリズム	17
2.4.3	旋律分類への適用	18
2.5	実験	22
2.5.1	準備	22
2.5.2	評価方法	22
2.5.3	実験結果	23
2.5.4	考察	24

	3
2.6 結び	26
<b>第3章 確率過程に基づく複旋律分類</b>	<b>29</b>
3.1 前書き	29
3.2 旋律の表現	30
3.3 旋律特徴量	32
3.3.1 単旋律における特徴量	32
3.3.2 複旋律における特徴量	33
3.3.3 本稿で提案する旋律特徴量	33
3.4 マルコフ過程とマルコフモデル	34
3.4.1 マルコフ過程	34
3.4.2 マルコフモデルによる文書分類	35
3.4.3 旋律分類への適用	36
3.5 実験	39
3.5.1 準備	39
3.5.2 評価方法	40
3.5.3 実験結果	40
3.5.4 考察	41
3.6 結び	44
<b>第4章 楽曲進行の推定による複旋律分類</b>	<b>49</b>
4.1 前書き	49
4.2 旋律の表現	50
4.3 旋律特徴量	50
4.3.1 単旋律における特徴量	50
4.3.2 複旋律における特徴量	51
4.3.3 本稿で提案する旋律特徴量	52
4.4 隠れマルコフモデル	53
4.4.1 隠れマルコフモデル	53
4.4.2 旋律分類への適用	54
4.5 実験	58
4.5.1 準備	58
4.5.2 評価方法	59
4.5.3 実験結果	59
4.5.4 考察	60
4.6 結び	63

第 5 章 結論

# 第1章 序論

## 1.1 扱う問題

今日において、コンサートや CD, DVD から、携帯電話や iPod に代表されるような携帯音楽再生機の普及、また作曲、編曲から著作権の問題など様々な分野において音楽を取り扱われている。特に、インターネット等の普及により、世界規模での情報が簡単に手に入れることが出来るようになった。音楽分野では、MIDI 等の記述式楽曲データの取り扱いは、多くの利用法がなされている。しかしその一方で、それらの情報の正確性の判断が難しくなっている。そこで、作曲家やジャンルといったユーザが求める、また判別しやすい形式で情報を得るために旋律分類に関する技術を確認することが本研究の目的である。

本研究では、確率モデルに基づく旋律分類を行う。将来、楽曲データのアーカイブ化などが予想される。大量に存在する楽曲データのアーカイブ化などでは、人手による判断では非常に困難であり、機械的に計算が出来るようになることは、将来的に非常に有用である。

楽曲情報・旋律情報を、一般的なニュース記事等の比較的情報量の多い文書集合と比較すると、1つの旋律から得られる特徴量の組み合わせは限定的な大きさである。旋律分類を行うにあたり注意しなければならない点は、この限定的な大きさのなかで、トリルなどノイズとしての要素を含む音が多く発生した場合に、分類正解率が低下する可能性がある。そこでテーマとなる要素を持った特徴を捉えることが出来れば、さらなる分類正解率の向上に繋がる。

## 1.2 背景と関連研究

楽曲を計算機で扱うための表現方法として、これまでに 信号レベル (CD 等でのアナログ的变化を表現)、操作レベル (MIDI など、輝度・時間のゆれを表現)、および楽譜レベル (表現・解釈の多様性を表現) が存在する。信号レベルにおける波形データのようなアナログ的な表現から、ある時刻における音の集合を抽出することは可能である。しかし、音楽理論におけるような情報、例えば拍子や音階、調性記号などの要素は、直接的な記述がないために情報として得ることは難しい。

旋律情報の表現は，大きく分けて単旋律 (*Monophony*) と複旋律 (*Polyphony*) の2つの形式に分けられる．単旋律とは，一般に主旋律を指す音の並びとなり，主旋律はメロディであると言い換えることも可能である．それに対し複旋律とは，一般にピアノ曲の場合なら両手鍵盤の音の並び，オーケストラなど複数の楽器による演奏では，それぞれの楽器ごとのパートを指す．しかし，計算機がこれらの情報を得るには，一定の形式に変換する必要がある．いずれの旋律情報に対しても，特徴的な旋律を捉えることが出来れば，分類を行う際に正解が得られやすい．

音楽情報処理が抱える旋律情報の定量化などの問題は，他の人工知能関連のテーマと同様に，分類問題に帰着させて定式化し，情報検索手法や機械学習手法により，自動計算するのが現実的である．

旋律特徴量とは，旋律情報から旋律の有する意味表現を考慮するために用いる．これまでに提案された特徴量記述では，旋律輪郭 (*Melody Contour*) が代表的である [1, 7, 18] が，この手法では，移調に強いという性質を持つが，その代わりに雑音に反応しやすく変化の激しい曲では，うまく性質を捉えられない側面もある．また，機械学習で扱うためには，それぞれの手法で必要となるアイテム集合が異なり，これまでに有効な旋律特徴量を提案されていない．

## 1.3 必要とされる機能

本研究では，旋律特徴量を用いてジャンルや作曲者などのテーマに振り分ける方法を用いることで旋律分類を実現する．この技術を実現するために，単旋律情報を用いた場合の分類と複旋律情報を用いた場合の分類について評価する．また分類結果より自動的に与えられるコードの中身についても評価する．

### 1.3.1 単旋律情報に関する分類

これまでの研究においては，主に輪郭情報を利用した分類が多く取り扱われてきたが，この特徴量は，移調に強いという特徴を持つが，同時にノイズなどに大きく反応してしまう欠点を持ち合わせている．そこで本研究では，輪郭情報を拡張した手法について検証している．また同時に，スペクトラム情報についても輪郭情報の特徴を取り入れた新しい形式の特徴量について検証する．

少量のラベル付旋律集合をテーマとみなし，多量の未知旋律情報を用いて分類器を学習させ正しく分類できるかを検証する．

### 1.3.2 複旋律情報に関する分類

単旋律の場合と大きく異なる点は、特徴量の形式である。単旋律では同時に発生する音が1音と極めて限定的な状態であるのに対し、複旋律では、同時に複数の音、場合によっては非常に数多くの音が発生することが考えられる。そのため、どのような形式の旋律特徴量を扱うことで、旋律に含まれる情報を取り入れることが可能となるかが問題となる。

そこで本研究では、少量のラベル付旋律集合をテーマとみなし、多量の未知旋律情報を用いてテーマから作成したマルコフモデル (*Markov Model*) に基づいて分類を行う。

マルコフモデルを用いた計算においては状態遷移が重要な要素となるが、旋律情報における状態そのものの定義がされていないことが大きな問題となる。また、マルコフモデルで扱う状態を定義でき、分類計算の手法が適用できたとしても、実際の旋律情報を扱う際には、状態遷移の観測は非常に困難である。

そこで本研究では、少量のラベル付旋律集合をテーマとみなし、多量の未知旋律情報を用いてテーマから作成した隠れマルコフモデル (*Hidden Markov Model*) に基づいて分類計算を行い、正しく分類できるかを検証する。隠れマルコフモデルとは、実際に観測された出力から状態列を推定する手法である。しかし、実際の正解とは異なる状態列を得る可能性が残る。

### 1.3.3 旋律情報間の整合性

旋律分類を行う際に重要となる分類方法は、ラベル付旋律集合をジャンルや作曲家と言ったラベルとし、そのいずれかのラベルに振り分けを行う教師あり機械学習の手法をとる。その際に、ラベルとどの程度似ているかという点が重要になるが、マルコフモデル、あるいは隠れマルコフモデルのような状態列や観測値を用いた確率モデルの場合、実際には近似したラベルの楽曲にもかかわらず、分類したい旋律に、ラベルと一致する状態列や観測値がないために、確率値が0になってしまう0確率問題が発生する可能性がある。そこで本研究では、2つの特徴量の間で一定の尺度を与え、どの程度近似しているかを数値的に表現する類似度を導入する。

### 1.3.4 複旋律情報からのコード推定

曲を推定するに当たり、曲におけるコード (*Chord*) 情報は重要な要素を持っている。音楽理論におけるコードとは、2つ以上の音の積み重ねを指し、その構成され

た音の響きが，コードが現れた時点で最も適した音を表現している．コードの割当は，多くは人手によるものが多く，これを機械的に割り当てるためには，コードを認識するための入力情報と，入力に対する判別するための機能，そして判別結果に対する出力などが求められる．

そこで本研究では，隠れマルコフモデルによって与えられたコード進行が正しく付与されたか検証する．

## 1.4 論文の構成

本論文は，前節の問題に対するひとつの解を以下のような構成で論じる．

次章においては，少量のラベル付き楽曲から，EM アルゴリズムを用いて，ラベルなし楽曲を分類する手法を提案する．主なアイデアは，各楽曲から特徴量を抽出し，単純ベイズ法による学習過程に基づくことにある．なお，本内容については *IEEE Computer Software and Application Conference (COMPSAC)* 等において発表している．

第3章では，複旋律のラベルなし楽曲を，ラベル付き楽曲に分類することを提案する．主なアイデアは，各小節から和音の特徴量を抽出し，和音間の類似性を考慮しつつ，マルコフモデルに基づいて分類基準を設定することにある．提案手法は，小節の長さごとに実験で検証され，高い正解率を得ることを示す．なお，本内容については *7th Int. Conf. on Intelligent Data Eng. and Automated Learning (IDEAL)* 等において発表している．

第4章では，隠れマルコフモデルを用いた複旋律分類を提案する．小節ごとに特徴量（ピッチスペクトル）を出力値とみなして隠れ状態を最尤推定する．同時に，平均律コード（和音）を推定する手法も検討している．なお，本内容についてはデータ工学ワークショップ (DEWS) において発表している．

第5章では，本論文のまとめ，また本論文で扱えなかった課題について言及する．

## 1.5 発表論文

### 1.5.1 論文誌

1. 吉原幸輝，三浦孝夫: 確率過程に基づく複旋律分類, 日本データベース学会 Letters (*DBSJ Letters*), Vol.5, No.1, pp.137-140, 2006.

### 1.5.2 口頭発表

1. 吉原幸輝, 三浦孝夫: EM アルゴリズムを用いた旋律分類, データ工学ワークショップ (DEWS), 電子情報通信学会データ工学研究会, 2005.
2. Yoshihara, Y., Miura, T.: Classifying Melodies by Using EM Algorithm, *IEEE Computer Software and Application Conference (COMPSAC)*, pp.204-210, 2005.
3. 吉原幸輝, 三浦孝夫: 確率過程に基づく複旋律分類, データ工学ワークショップ (DEWS), 電子情報通信学会データ工学研究会, 2006.
4. Uejima, H., Miura, T.: Classifying Polyphony Music Based on Markov Model, *7th Int. Conf. on Intelligent Data Eng. and Automated Learning (IDEAL)*, pp.697-706, 2006.
5. 吉原幸輝, 三浦孝夫, 塩谷勇: 楽曲進行の推定による複旋律分類, データ工学ワークショップ (DEWS), 電子情報通信学会データ工学研究会, 2007.

## 第2章 EM アルゴリズムを用いた旋律分類

### 2.1 前書き

本稿では、複数のテーマ(単旋律)から特徴値を抽出し、未知の旋律を分類していずれかのテーマに帰着させる手法(旋律の自動分類)を論じる。これらは、本来手書き文字認識や手書き絵による類似検索と同様に、内容検索を目的とし、曲名や作品番号といったメタ情報を仮定しない。

旋律は主要な印象を与える役割を有しており、この機能を用いた応用は広範囲に想定できる。例えば、音楽作業(作曲・編曲)に対して直接的な支援を行うことや、同じクラスの曲を検索し著作権に抵触しそうな候補の絞込みを行うことができる。実際、音楽における著作権管理では、歌詞と異なって、旋律で判断することが多い<sup>1</sup>。また、インターネットを介して得た音楽コンテンツの自動管理においても旋律による手がかりを与えることができる。

音楽情報処理分野の研究は60年代から開始され、マルチメディア処理技術(信号処理からメタ情報による検索まで)や情報検索理論との関連(N グラム, ベクトル空間モデル, 検索エンジン等)が論じられてきた。

他の情報検索分野と比較して、音楽分野での分類問題は特に難しい点がある。まず、音楽情報をモデル化するのにデータモデリング機能が著しく欠けていることが挙げられる。第2に旋律の特定が難しい。実際、旋律の特徴量を定量的に定義することができず、改ざんによる旋律の同定を扱えことさえできない。第3に、旋律の抽出方法が不明であることも重要である。通常、テーマ(主旋律)が中心になるが、編曲の過程で和音、楽器・音色などが付与される。テーマである旨の明示的な記述表現もない。第4に、旋律の類似性の判定(何が答えか?)が明確に定義できず、クラス所属も確率やファジーメンバーシップによるのか、ランキング表現がよいのかが定まらない[18]。最後に、効率よいデータ管理・データ操作に関して、どのような技術が必要かが不明であることも重要である[15]。

<sup>1</sup>歌詞と切り離れた、音楽そのもの「旋律、和音、リズム、形式の4要素が一体となった楽曲」(東京地方裁判所昭和43年5月13日)も独立に著作物となり、保護の対象となる。

このような音楽情報処理の問題は、他の人工知能関連のテーマと同様に、分類問題に帰着させて定式化し、情報検索手法や機械学習手法により、自動計算するのが現実的である。

機械学習手法を用いた帰納推論では、分類クラス(ラベル)が明記された幾つかの訓練データから一般的なパターン(分類器)を抽出し、これを利用して未知データを分類する。正確に分類するには大量の訓練データを必要とするが、ラベルをつける過程は通常人間が行うため、時間がかかり主観的で信頼性にかけたものになりがちである[11]。インターネットを介して得るデジタル音楽のラベルの多くは誤っているか無記名であり、ラベルを正確につける手間は困難な状況になっている。

近年これに対応するために、不完全データによる分類器生成手法(ブートストラップ手法)が提案され、とくにEM アルゴリズムが代表的である[27]。ここでは少量のラベルつき訓練データとラベル無し訓練データから分類器が構成される。

この方式の利点は、少量の訓練データから自己増殖的に学習するために未知データを活用できることにある。すなわち、インターネット上に点在する音楽データを効果的に学習できることにあり、以下で実験によりその有効性を検証する。

本研究の目的は、このような少量のラベル付き訓練データから自動分類を行う旋律分類手法の提案にある。ここでは分類器の生成自体が目的ではなく、ラベル無しデータの推定精度を確かめることにある。本研究では、旋律の自動分類の精度を向上するため、旋律(楽譜)から特徴量を抽出し、機械学習手法を用いて分類に利用する。具体的には、旋律特徴量を基に単純ベイズ手法で分類し、さらにEM アルゴリズムによって計算途中の分類結果を利用する。

本稿では、音楽に関する基本知識[26]、および情報検索技術の基本知識を仮定する[5, 29]。

第2節で、準備と必要な定義を述べ、第3節で音楽情報の特徴量を要約し、本研究で扱う特徴量を定義する。第4章では、主として分類に利用する単純ベイズ法およびEM アルゴリズムの適用方法を示し、この特徴や利点を述べる。第5節では実験結果を述べ、本手法の有効性を示す。

## 2.2 旋律の表現

楽曲は複数の音から構成される[26]。ピッチ(pitch)とは音の調子、即ち音の高さを表す。これは周波数で規定され、代表的に記号で表現される。例えば、A(ラ)の音は440Hzであり、その倍の高さの音を1オクターブ(octave)という。例えば880HzはAの1オクターブ高いaを表す<sup>2</sup>。ピッチを音(tone)とも呼ぶ。

<sup>2</sup>以下では1オクターブ高いピッチを小文字で表す。

音程(interval) はピッチ間の距離であり, 1 オクターブ離れた音程を 12 等分した単位を半音(semitone) という. 例えば A から a までは A, A#, B, C, C#, D, D#, E, F, F#, G, G# と表す. G#, A#, C#, D#, F# はそれぞれ Ab, Bb, Db, Eb, Gb と表す. 全音階的(diatonic) な音の並びとは, A, B, C, D, E, F, G, a または C, D, E, F, G, A, B, c のように, 半音がそれぞれ 2, 1, 2, 2, 1, 2, 2 または 2, 2, 1, 2, 2, 2, 1 ずつ離れた配置を言う. 前者を短調(minor) 後者を長調(major) という. また A, C をそれぞれの主音(key), +7 音を属音(dominant) という. 全音階的でない並びを無調的(atonal) という. 移調(transposition) とは, 主音を短調・長調の並びを保ったまま相対的に移動することをいう.

各ピッチには長さ(duration) がある. 旋律(melody) とは(長さを持つ)ピッチの順次配置である. 和音(chord) は複数のピッチの並列配置であり, これらは同時に演奏されることを想定している. 音楽の意図する主要な旋律ではない部分を伴奏(accompaniment), 伴奏の付いた音楽を多声音楽(polyphony), 伴奏が無い音楽を単旋律(monophony) という.

五線譜(five horizontal lines あるいは score) は, 音の高さと長さを記入した記号(音符) を順次的あるいは並列的に記述した表現である. 旋律は音符の並びで, 和音は音符の重なりで記述される. 音符の長さの合計を拍(beat) という. 小節(bar) とは同じ拍数を持つ(五線譜の)部分記述であり, 拍子(time signature) によって小節当たりの拍数と拍質を表す. 特に, リズム(rhythm) とは拍の規則的な並びをいう.

音楽を表現するためには, 多くの表情つけがなされている. これらは大きく 4 つに分けられ, 速度・強弱に関するもの, 発想に基づくもの, 奏法によるもの, 装飾音に関するものから成り, 五線譜上に直接記載される.

これらを計算機で表現するには, これまで信号レベル(CD 等でのアナログ的变化を表現), 操作レベル(MIDI など, 輝度・時間のゆれを表現) 及び楽譜レベル(表現・解釈の多様性を表現) でなされてきた.

## 2.3 旋律特徴量

旋律を分類するためには, 当該音楽の有する特徴値の表現を考慮する必要がある. 旋律を分類するために指定される条件は, 不完全である場合が多く, その大半は部分的である.

本研究では旋律から特徴量を抽出し, ベクタ空間モデル [5, 29] を用いて特徴項目ベクトル空間へ写像する. 各次元は項目に対応するため, どの旋律も空間内の点で表せる. 旋律質問の解はベクトル空間内の類似した点に対するランクつけで

得られる [20] . 空間内では伝統的な余弦質問で定義する . 即ち  $\cos \theta = \frac{X \cdot Y}{|X| |Y|}$  を計算し , 1.0 に近いほど類似していると判断する . ここで  $X, Y$  は空間ベクトルを ,  $X \cdot Y$  は内積を ,  $|X|$  はベクトル長を表す .

音楽情報を分類するためには , クラスがあらかじめ判明している旋律 (ラベル付き訓練データ) と未知旋律の関連を評価する尺度が必要である .

本稿ではピアノ曲に限定して考察する . ピアノ曲では通常主旋律を奏するのは右手である . 右手に対応した第 1 声に限定し , 和声があった場合は (高音とは小指や薬指になるので限定しにくいことから) 和声記述の先頭音を旋律音と見なす発見的な方法を取る .

### 2.3.1 ピッチ輪郭

これまで提案された旋律情報のための特徴量記述では , 旋律輪郭 (Melody Contour) が代表的である [1, 7, 18] . また , 近年は楽譜化技法として , XML に基づくものや五線譜利用による抽象化記述がある . 旋律の輪郭情報のうち , ピッチ輪郭 (pitch contour) 表現は多くの研究で検討されている . ピッチ輪郭表現とは , 単旋律にだけ有効であり , 直前の音と比べてピッチが相対的に高い , 低い , 同じという状態をそれぞれ文字  $U, D, S$  で表し , 輪郭を文字列として記述する<sup>3</sup> . この表現は , ピッチに対して相対的なので移調に強いが , 雑音に反応して文字列が変わりやすく , 旋律検索は不完全文字列一致処理の実行を意味する .

#### カエルの歌 ピッチ輪郭の例

以下では , "カエルの歌" を統一例として考察する . 五線譜は図 2.1 で示されるが , 2 段譜の右手パートに主旋律が生じている . このピッチ輪郭は -UUU DDD UUUU DDD DSSS SSUSUSUS DDD である . あるいは , 長さを無視することはピッチが同じかぎり意味をなさないと考えることに相当するので , 記号  $S$  を取り去り , -UUU DDD UUUU DDD D UUU DDD と表すこともある .

しかし , ピッチ輪郭方式は数多くの問題を含む [17] . 大規模楽曲の記述から主旋律を抽出することは難しいため , 多声音楽から各声を旋律と見なした輪郭を記述し , これらの結合から主旋律を想定する , という組み合わせ的探索を行わざるを得ない [2, 3] . 旋律記述の記述に拍子・音符長を含まないため , 判別のため多くの情報を必要とする . 例えば -SSD USSD は運命交響曲 (Beethoven) も雪山賛歌 あ

<sup>3</sup>開始文字は相対位置を定めることができないので - で表す .



図 2.1: カエルの歌五線譜

るいは Oh! My Darling, Clementine! (米国民謡) も表すためより多くの情報を必要とする。また，不完全一致検索手続きは多項式時間でよいことが知られるものの複雑で，(全走査を除けば)<sup>2</sup> 次記憶域上での処理効率を向上させる方法は知られていない。輪郭抽出には雑音除去のための”しきい値”の設定が容易ではなく，生成される輪郭を規定する。

### 2.3.2 ピッチスペクトル

音楽データベース情報は，特定されるべき楽曲であるから五線譜情報を有するほうが有利である。実際，これらは高度の抽象性を有し，厳格な指示(五線譜上の記号による指示)に忠実に従っている。不完全な旋律指定(不正確・欠落・旋律やリズムの近似，移調・長単調の変更，作為的な表情付け，こぶし，ずれ，ポルタメントといった揺らぎ)に対して，楽譜記述を用いれば，リズム・音色・発想・速度・奏法・強弱・拍子などの観点で問題を局所化する。反面，移調・装飾音・揺らぎという問題については，楽譜化では容易には解決できない。

旋律が五線譜・楽譜で記述されるとき，小節単位にピッチスペクトル(pitch spectrum)を導入し特徴量とすることができる[10]。ピッチスペクトルとは，小節内に生じる各音符を長さで集積しそのヒストグラムをとるもので，以下では4オクターブ  $12 \times 4 = 48$  音を対象に，48次元ベクトルを生成する<sup>4</sup>。小節ごとにピッチスペクトルを算出し，これを対象楽曲の規定小節数だけ用意する。このアイデアは信号処理では提案されているが，楽譜レベルで特徴量化するというアプローチはない[22]。

#### カエルの歌 ピッチスペクトルの例

カエルの歌(2声)から抽出した旋律(第1声)に対する16小節分のピッチスペクトル(図 2.2)を示す。

<sup>4</sup>倍精度実数で表せば  $8 \times 48 = 384$  バイト長となる。

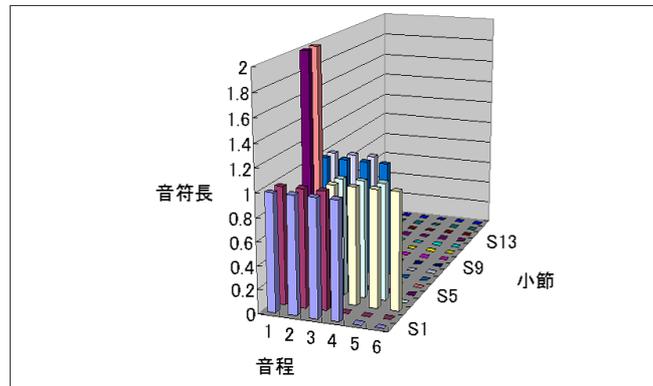


図 2.2: カエルの歌ピッチスペクトル

ピッチスペクトルは各ピッチを長さで合計したヒストグラムであり、(ギターコードのような) 小節全体に和音を指定したものと異なる。ピッチ輪郭とは異なって、小節単位の特徴を集約しているため、旋律の輪郭を表してはいない。むしろ、小節間の変化を記述しているため、旋律の時系列表現とも異なる。ピッチスペクトルは不完全な旋律に生じる問題を解消することができる。実際、装飾音・旋律・長短調・揺らぎがピッチスペクトルで局所化する。

ピッチスペクトルの問題は 移調に変動しやすいこと、および長調・短調の変化に対応しづらいことであることが知られている [10]。前者については、旋律から主音(キー)を決定しテーマへの移調を行えばよい。しかし後者の問題については対応が難しく、より高度な対応が必要である。

### 2.3.3 本稿で提案する旋律特徴量

本稿では、旋律分類を行うための特徴量を、ピッチ輪郭およびピッチスペクトルに加え、平滑化ピッチ輪郭 (Mean Contour)、ピッチ細分輪郭 (Pitch Fine Contour) および輪郭スペクトル (Contour Spectrum) を導入し、EM アルゴリズムによってどの程度まで分類精度を向上できるかを論じる。

ピッチ輪郭手法には幾つかの重要な欠点がある [7]。その第 1 は、旋律内のピッチの微妙な変化 (雑音) に影響を受け易いことにある。例えばトリルのような細かなピッチの変動は、印象をほとんど変えないにもかかわらず、ピッチ輪郭では大きな差となる。そこで、平均フィルタ (Mean Filter) と同様に、各音を前後と合わせて平均化し、この結果に対してピッチ輪郭で表現する手法を、平滑化ピッチ輪郭 (Mean Contour) と定義する。

## カエルの歌 ピッチ輪郭の例

カエルの歌を平滑化すると，第 1 小節最終音(“ファ”)がその前後と平均化され“ミ”に代わる．同様に第 3 小節最終音(“ラ”が“ソ”に)に代わり，その結果，この平滑化ピッチ輪郭は-UU DD UU DD D UUU DDD である．

ピッチ輪郭手法の第 2 の問題は，旋律内のピッチの変化を上昇(U)または下降(D)でしか表現できないことにある [7]．滑らかな音階進行もオクターブ移動のような激しい進行もピッチ輪郭では差になって生じない．そこで，ピッチの移動幅(半音を 1 として)を付加しピッチ輪郭を拡張表現する手法を，ピッチ細分輪郭(Pitch Fine Contour)と定義する．

## カエルの歌 ピッチ細分輪郭の例

カエルの歌をピッチ細分輪郭で表現すると次のようになる．

-U2U2U2 D1D2D2 U4U1U2U2 D2D2D1 D4 U2U2U1 D1D2D2

しかし，この方法では，明らかに雑音の影響を(ピッチ輪郭以上に)受けやすい．このため，ピッチスペクトルと同様に小節ごとにスペクトルを取りベクトル化する．これを輪郭スペクトル(Contour Spectrum)という．ピッチ輪郭は長さ情報を考慮していないため，ベクトル長さは 1 に正規化する．輪郭スペクトルは上昇下降それぞれ 24 までと仮定して 48 次元ベクトルで表す．例えば D2 が 3 回生じるとき，D2 次元に 3 を設定し，最後に長さが 1 になるように調整する．以下では D2 が 3 回生じるときに D2:3 と表現する．

## カエルの歌 ピッチスペクトルの例

カエルの歌を輪郭スペクトルで表現すると 8 個のベクトル列になる．

{ U2:3 }, { D1:1, D2:2 }, { U1:1, U2:2, U4:1 }, { D1:1, D2:2 },  
{ D4:1 }, { }, { U2:2, U1:1 }, { D1:1, D2:2 }

輪郭スペクトルは，ピッチ輪郭の利点をすべて有している．更に雑音にも強い．実際，ピッチの移動幅が多少異なってもベクトルの類似性(余弦値)は消えない(0 にはならない)．小節単位で同期を取るために，対応が取れない部分があってもそれ以降に影響を与えないという意味で，ピッチ輪郭の問題に対応している．

## 2.4 単純ベイズ法とEMアルゴリズム

以下では、出典の知られている旋律  $d$  をラベル付き訓練データとして、未知旋律をいずれかのクラスに分類する。本研究ではEM アルゴリズムと単純ベイズ分類を組合わせた手法を用いる [12]。  $d$  は、特徴量  $w_j$  を用いて多重集合  $\{w_1, \dots, w_m\}$  として表されると仮定する。

### 2.4.1 単純ベイズ法による文書分類

旋律  $d$  に、クラス集合  $C = \{c_1, \dots, c_n\}$  のいずれかの値を割り当てる (分類する) ことを考える。ベイズ規則による分類は、 $d$  がクラス  $c$  に属する確率  $P(c|d)$  の確率分布を求めることである。排他的な分類の場合、最大事後確率をとるクラス  $c \in d$  を分類することで分類エラーが最小になると考える。ここでクラスラベルは次のように定める:  $c_k = \operatorname{argmax}_{c \in C} P(c|d)$ 。ベイズ規則は以下のように与える:  $P(c|d) = P(c) \times \frac{P(d|c)}{P(d)}$ 。すなわち、ベイズ規則での分類規則生成 (訓練) は訓練データ集合から、確率分布  $P(c)$ ,  $P(d)$ ,  $P(d|c)$  を推定することである。

しかし、旋律  $d = \{w_1, \dots, w_m\}$  に関して  $P(d|c)$  や  $P(d)$  の推定が問題であるため、一般に、特徴量  $w_j$  の出現は、統計的に他の  $w_i$  の出現とは独立であるという仮定をおき、各旋律を特徴量の集合とみなす単純ベイズ手法を使うことが多い。単純ベイズでは、 $P(d|c)$  を以下の形式に分解して考える。  $P(d|c) = \prod_{i=1}^{|d|} P(w_j|c)$ 。これにより、排他的分類の場合ではベイズ規則は以下のように書くことができる。

$$P(c|d) = P(c) \times \prod_{i=1}^{|d|} P(w_j|c) \quad (2.1)$$

また、ここでは旋律内での各特徴量は、その定義に従って与えられる。ベイズ推定手法は特に文書分類で有効であることが知られている [28, 30]。

### 2.4.2 EM アルゴリズム

EM アルゴリズムとは、データの欠損部分 (直接観測できない確率変数) を最尤推定により求める、欠損部が分かればその形は単純で解析的に表現できるという仮定を置く。

Expectation (期待値), Maximization (最大化) は、それぞれ欠損値の推定, 期待値を得る過程を与えるパラメタの推定に対応している。不完全な観測データ  $x_1, \dots, x_N$  が、モデル  $P_\theta(x)$  から得られたとする。EM アルゴリズムとは、この不完全な観測データ  $x_1, \dots, x_N$  から、未知のパラメータ  $\theta$  の値を推定することである。現時点で

の  $\theta$  を使用して、条件付確率モデルから、完全データにおいてのサンプル数、期待値を求める (E ステップ)、続いて、ここで求めたサンプルから期待値を最大化するパラメタ  $\hat{\theta}$  を求める (M ステップ)。この E ステップと M ステップを繰り返すことにより、モデルの対数尤度を最大化するパラメータを求める。

EM アルゴリズムを分類操作に適用するアルゴリズムは以下のようになる。

1. 入力：ラベル付旋律集合，ラベルなし旋律集合
2. ラベル付旋律集合から，単純ベイズ分類規則  $\hat{\theta}$  を生成
3. 以下のステップを一定回数，またはパラメタが収束するまで繰り返す
  - (E-step) 現在の分類規則  $\hat{\theta}$  を使用してラベルなし旋律のクラス所属確率 ( $P(c_j|d_i; \hat{\theta})$ ) を推定する
  - (M-step) 推定された事後確率 (分類結果) を利用して，分類規則  $\hat{\theta} = P(D|\theta)P(\theta)$  を再度生成する。
4. 出力：分類規則  $\hat{\theta}$  を用いて，各ラベル無し旋律のラベル  $c_k$  は次のように定める:  $c_k = \operatorname{argmax}_{c \in C} P(c|d)$

EM アルゴリズムでは各繰り返しステップで尤度が単調増加するため，必ず解を有する。反面，収束が遅く局所解に至る可能性があるため，初期値の選定が難しい [27]。

### 2.4.3 旋律分類への適用

様々な特徴量を使用することで単純ベイズ法に基づく EM アルゴリズムを旋律分類に適用する方法を説明する。

$C = \{c_1, \dots, c_m\}$  はクラス集合を表す。ラベル付き旋律  $e_c$  はいずれかのクラスに含まれる  $c \in C$ 。  $D$  はすべての訓練データを表し， $w_i$  はデータ内の各特徴量を表す。また  $N(w_i, d_j)$  は旋律  $d_j$  における特徴量  $w_i$  の特徴値である。さらに， $P(c_k|d_j)$  は前述の旋律  $d_j$  がクラス  $c_k$  に属する確率を表し，ラベル付き訓練データに関しては，そのラベル付けられたクラス  $c_m$  においては， $P(c_m|d_j) = 1$  であり，それ以外のクラスに対しては 0 をとる。これに対して，ラベルなし訓練データは，当初は全クラスに対して 0 であるが，最初は通常のベイズ分類により，その後は EM アルゴリズムにより，徐々に適切な値へと更新される。

$$P(w_i|c_k) = \frac{1 + \sum_{j=1}^{|D|} N(w_i, d_j)P(c_k|d_j)}{|V| + \sum_{i=1}^{|V|} \sum_{j=1}^{|D|} N(w_i, d_j)P(c_k|d_j)} \quad (2.2)$$





図 2.3: Morzart KV265



図 2.4: Beethoven Symphony No.9, Op.125

(1) 単純ベイズ法によりクラス割り当てを行う。定義より  $P(A|d_1) = P(B|d_2) = 1$ ,  $P(B|d_1) = P(A|d_2) = 0$  and  $P(A|d_3) = P(B|d_3) = 0$ .

である。ここでは Bayes 定理を用いて

$$P(C) \times P(d_3|C).$$

を最大化する  $C$  を求めたい。この式は  $P(C|d_3) \times P(d_3)$ <sup>5</sup> を意味している。

$D = L_1 \cup L_2$  には 3 件のデータが含まれ、クラス A,B にはそれぞれ 1 件しか含まれていないから、 $P(A) = P(B) = 1/3$  である。

単純 Bayes の仮定から  $P(d_3|C) = P(U|C) \times P(D|C)$  である。ここで  $P(A|d_3)$ ,  $P(B|d_3)$  をそれぞれ推定する。

$$P(A) \times P(d_3|A) = P(A) \times P(U|A) \times P(D|A)$$

$$= \frac{1}{3} \times \frac{7}{13} \times \frac{6}{13} = 0.083$$

$$P(B) \times P(d_3|B) = P(B) \times P(U|B) \times P(D|B)$$

melody	(U,D)	U+D
$d_1$	7, 6	13
$d_2$	2, 1	3
$d_3$	4, 5	9

class	$P(U *)$	$P(D *)$
A	7/13	6/13
B	2/3	1/3

<sup>5</sup> $P(d_3) = P(U) \times P(D) = (13/25) \times (12/25)$

$$= \frac{1}{3} \times \frac{2}{3} \times \frac{1}{3} = 0.073$$

この結果、 $P(C|d_3)$  を最大化する  $C$  は  $A$  であり、“歓喜の歌” は “カエルの歌” に割り当てられた。

旋律  $d_3$  が  $A$  に割り当てられた結果、 $P(A) = 2/3$ ,  $P(B) = 1/3$  に変わる。

また、 $d_3$  のクラスが割り当てられた結果条件確率  $P(U|A)$ ,  $P(D|A)$  が変わる。

class	$P(U *)$	$P(D *)$
A	11/22	11/22
B	2/3	1/3

更に  $d_3$  のクラス所属確率  $P(C|d_3)$  は  $P(C|d_3) = P(C) \times P(d_3|C)/P(d_3)$  で求められ、それぞれ次の値となる。

$$P(A|d_3) = P(A) \times P(d_3|A)/P(d_3) = \frac{2}{3} \times \frac{11}{22} \times \frac{\frac{11}{22}}{\frac{13}{25} \times \frac{12}{25}}$$

$$P(B|d_3) = P(B) \times P(d_3|B)/P(d_3) = \frac{1}{3} \times \frac{2}{3} \times \frac{\frac{1}{3}}{\frac{13}{25} \times \frac{12}{25}}$$

(2) 新たに定まった確率値を用いて、再度  $d_3$  の所属確率を求める。 $P(C|d_3)$  を最大化する  $C$  を求めるため、次式を計算する。

$$P(A) \times P(d_3|A) = P(A) \times P(U|A) \times P(D|A)$$

$$= \frac{2}{3} \times \frac{11}{22} \times \frac{11}{22} = 0.167$$

$$P(B) \times P(d_3|B) = P(B) \times P(U|B) \times P(D|B)$$

$$= \frac{1}{3} \times \frac{2}{3} \times \frac{1}{3} = 0.073$$

この結果、 $P(C|d_3)$  を最大化する  $C$  は  $A$  のままであり、“歓喜の歌” は “カエルの歌” に割り当てられた状態を保つ。 $d_3$  の所属確率を求めるための状況は先と変更が無いため、ここで繰り返しを停止する。

後述するように、“カエルの歌” と “歓喜の歌” は、ピッチ輪郭を用いた類似度は 90.3% であり、かなりの確度で分類したことになる。

## 2.5 実験

以下の実験では幾つかの変奏曲を扱う。変奏曲は1つのテーマと複数の変奏曲から構成されるため、ラベル付き訓練データはテーマの数だけあると考える。これ以外はラベル無しデータとみなし、旋律分類の対象とする。

### 2.5.1 準備

本稿では、ピアノ変奏曲(単旋律)を用いる。使用する楽曲は、Mozart キラキラ星変奏曲 K.265, Schubert 即興曲 Op.142-3, および Beethoven トルコ行進曲による6つの変奏曲 Op.76 の3つの変奏曲であり、それぞれ主題(テーマ)と12曲、5曲、6曲の変奏(バリエーション)からなる。これらは MIDI データとして表現されているものを ABC フォーマット [21] に変換して実験に使用する。

まず最初に、すべての単旋律の ABC ファイルから特徴量を取り出す。特徴量は、曲の先頭4小節分の音を対象とする。取り出した特徴量を、ラベル付旋律集合とラベルなし旋律集合に分ける。本実験では、ラベル付旋律集合は変奏曲のテーマ(主旋律)とし、残りの変奏曲はラベルなし旋律集合とする。

また、ここで抽出する特徴量は、

- ピッチ輪郭：上昇(U)，下降(D)，同等(S)の3アイテム
- ピッチ細分輪郭：オクターブを6音とした，D6 から U6 の13アイテム
- 平滑化ピッチ輪郭：オクターブを6音とした，D6 から U6 の13アイテム
- 輪郭スペクトル：オクターブを6音とした，D6 から U6 の13アイテム
- ピッチスペクトル：半音を除いた7音+休符の8アイテム

とする。

### 2.5.2 評価方法

特徴値ごとに、2つの旋律の類似度について次のように定義する。

輪郭情報(pitch contour, mean contour, pitch fine contour)の場合は、旋律の先頭4小節について、最終的にアイテムごとに集計された頻度ベクトルとなる。例えば pitch contour では U,D,S の個数からなる3次元ベクトルを得る。このとき旋律 E1,E2 の類似度を特徴値ベクトル V1,V2 の余弦と定義する。

スペクトル情報 (pitch spectrum, contour spectrum) の場合は, 各小節ごとにスペクトル (ベクトル) が生成される. テーマおよび変奏曲 T, E のスペクトル列  $\langle A1, A2, A3, A4 \rangle$ ,  $\langle B1, B2, B3, B4 \rangle$  に対して小節ごとに余弦値を求め特徴値ベクトル  $V = \langle V1, V2, V3, V4 \rangle$  とする. ここで  $V1$  は  $A1, B1$  の2つのベクトルの余弦値を示す. このとき, 2つの旋律  $E1, E2$  の類似度を, 特徴値ベクトル  $V, W$  の余弦と定義する. 特にテーマ T と変奏曲 E の類似度は  $\langle 1, 1, 1, 1 \rangle$  と特徴値ベクトルの余弦として求められる.

本実験では, 単純ベイズ法だけを用いた分類と EM アルゴリズムを用いた分類による正解率の変化や, 特徴量の抽出方法による正解率の変化を比較することで, どの抽出方法がよいか, または EM アルゴリズムの繰り返し回数に関係するのかを検証する. 正解率は, 次式で定義される.

$$\frac{\text{類似度以上で正しく分類された総変奏曲数}}{\text{総変奏曲数}} \times 100(\%) \quad (2.5)$$

次に, ラベルなし旋律集合を分類する. 最初に持っている分類規則は, ラベル付旋律の特徴量とする. 4.2 で述べた方法で, この分類規則を用いて EM アルゴリズムを適用する. その分類規則から, ラベルなし旋律のクラス所属確率を, ラベル付旋律の特徴量とラベルなし旋律の特徴量から推定する (E-Step). また, 類似度 (余弦値) を求め, その値が 30%, 50%, 60%, 66.7% 以上の時, ラベル付旋律集合に分類されたとみなし, それ未満の場合は, 分類されたと考えない (正解率の対象ではない). 同じ変奏曲に分類された場合を正解とする. そして, 所属確率に基づいて, 分類規則を学習することで, 新しいラベル付旋律の特徴量を得る (M-Step).

これらの, E-Step と M-Step を一定回数繰り返し, その時の正解率を見る. 本実験では, 繰り返し回数を, 0回 (単純ベイズ法のみ), 5回, 10回, 15回, 20回, 25回の6パターンについて結果をまとめる.

### 2.5.3 実験結果

特徴量の抽出に, ピッチ輪郭を使用した際の正解率を表1に, ピッチ細分輪郭を使用した際の正解率を表2に, 平滑化ピッチ輪郭を使用した際の正解率を表3に, 輪郭スペクトルを用いた使用した際の正解率を表4に, ピッチスペクトルを用いた使用した際の正解率を表5に示す.

また EM アルゴリズムによる改善の割合を示す. 類似度 30% 以上の場合を図 2.5 に, 類似度 50% 以上の場合を図 2.6 に, 類似度 60% 以上の場合を図 2.7 に, 類似度 66.7% 以上の場合を図 2.8 に示す.

		EM 回数					
		0	5	10	15	20	25
類似度 (%)	30	34.8	34.8	52.2	52.2	52.2	52.2
	50	34.8	34.8	52.2	52.2	52.2	52.2
	60	34.8	34.8	47.8	47.8	47.8	47.8
	66.7	34.8	36.4	47.8	47.8	47.8	47.8

表 2.1: ピッチ輪郭 (pitch contour)

		EM 回数					
		0	5	10	15	20	25
類似度 (%)	30	39.1	31.8	31.8	31.8	31.8	31.8
	50	39.1	26.1	26.1	26.1	26.1	26.1
	60	31.8	26.1	26.1	26.1	26.1	26.1
	66.7	31.8	26.1	26.1	26.1	26.1	26.1

表 2.2: ピッチ細分輪郭 (pitch fine contour)

ピッチ細分輪郭，平滑化ピッチ輪郭，輪郭スペクトルを使用した結果は，単純ベイズ法のみでの計算で終えた時が正解率が一番よく，輪郭スペクトルを使用した結果では，類似度 30 % 以上の場合，47.8 % の正解率を示した．ピッチ輪郭とピッチスペクトルを使用した結果では，EM アルゴリズムを 10 回繰り返した場合が，一番正解率が向上し，ピッチスペクトルを使用した結果では，類似度 60 %，66.7 % 以上の場合，26.1 % も正解率が向上した．

#### 2.5.4 考察

結果から明らかのように，ピッチ輪郭とピッチスペクトルの場合において EM アルゴリズムによる学習方法は，結果を向上させることができる．実際，ピッチ輪郭に関しては，どの類似度の場合でも，EM アルゴリズムの繰り返しが 5 回から 10 回までの間に 15 % 前後の向上，ピッチスペクトルに関しては，30 % 以上の類似度の場合で 0 回から 5 回までの間に 4 % 前後の向上，50 % 以上の類似度の場合で 0 回から 5 回までの間に 12 % 前後の向上，60 % 以上の類似度の場合で 0 回から 5 回までの間に 20 % 強の向上，5 回から 10 回までの間に 4 % 前後の向上，66.7 % 以上の類似度の場合で 0 回から 5 回までの間に 25 % 強の向上が見られる．この理由として，1 つは，1 回 EM アルゴリズムの計算をした時点で，間違っ

		EM 回数					
		0	5	10	15	20	25
類似度 (%)	30	26.1	17.4	17.4	17.4	17.4	17.4
	50	26.1	17.4	17.4	17.4	17.4	17.4
	60	26.1	17.4	17.4	17.4	17.4	17.4
	66.7	26.1	17.4	17.4	17.4	17.4	17.4

表 2.3: 平滑化ピッチ輪郭 (mean contour)

		EM 回数					
		0	5	10	15	20	25
類似度 (%)	30	47.8	34.8	34.8	34.8	34.8	34.8
	50	39.1	34.8	34.8	34.8	34.8	34.8
	60	34.8	26.1	26.1	26.1	26.1	26.1
	66.7	26.1	21.8	21.8	21.8	21.8	21.8

表 2.4: 輪郭スペクトル (contour spectrum)

ラベルが付けられた変奏曲は，正解ラベルの所属確率と，誤ラベルの所属確率が大きくないことである．これにより，EM アルゴリズムを数回行った結果，正解のラベルが付けられている．もう1つは，正解ラベルであるにもかかわらず，類似度が低いために正解率の対象とならなかった変奏曲が，EM アルゴリズムの繰り返し計算によって，類似していくことになったと考えられる．

他方，他の特徴量は EM アルゴリズムでは効果的でない．この原因として，特徴値が細かいために，先頭4小節の特徴量では類似性を見つけるのが困難で，少ない特徴量の中で雑音がテーマ（主旋律）とマッチしてしまうと，誤ラベルの所属確率が大幅に増加してしまったことが考えられる．また，正解ラベルの所属確率と誤ラベルの所属確率の差が大きく，EM アルゴリズムを20回程度の繰り返しでは，差は縮まるもののラベルの変更までは至らないこともある．

繰り返し回数がある一定数を超えると正解率が低下する．この原因として，EM アルゴリズムで求める分類規則が推定のものであるため，初期値（本実験ではテーマ）が正解のラベルなし旋律と大きく異なると，その影響を繰り返しのたびに誤りを増大させていると考えられる．実際，例としてピッチ細分輪郭を用いた分類で，キラキラ星変奏曲テーマ，Schubert 即興曲テーマ（共にラベルつき訓練データ），ラベルなし旋律のキラキラ星変奏曲2番を比較する．

単純ベイズ法のみでの計算では，K.265 v2 は，キラキラ星変奏曲と正しく分類さ

		EM 回数					
		0	5	10	15	20	25
類似度 (%)	30	87	91.3	91.3	91.3	91.3	91.3
	50	78.3	91.3	91.3	87	87	87
	60	60.9	82.6	87	82.6	82.6	82.6
	66.7	52.2	78.3	78.3	69.6	65.2	65.2

表 2.5: ピッチスペクトル (pitch spectrum)

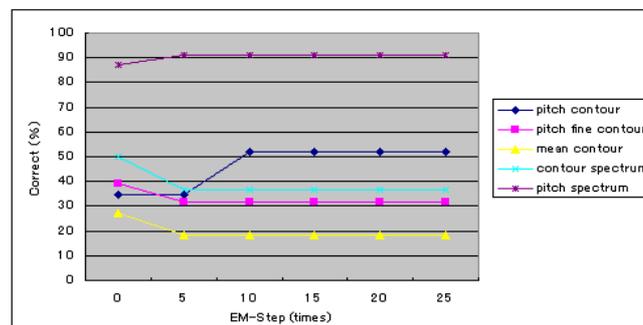


図 2.5: 類似度 30 % 以上

れた．ただ，EM アルゴリズムを 1 回計算すると，Schubert 即興曲に分類されてしまう．D1 : S の出現回数を見ると，K.265 テーマ はほぼ 1 : 1，Op.76 テーマと K.265 v2 はほぼ 1 : 2 であることが分かる．EM アルゴリズムによりこの形に従うように変更していく．したがって繰り返しの都度，誤分類をますます強めていく状況を生んでいる．

## 2.6 結び

本研究では，少量のラベル付き楽曲から大量のラベルなし楽曲を分類する方法を提案した．具体的には，5 つの特徴量の抽出方法と，単純ベイズ法と EM アルゴリズムを組み合わせることにより，分類精度が改善できるかを評価した．この結果，ピッチ輪郭とピッチスペクトルでは EM アルゴリズムの特性を上手く捉えており，分類精度を改善させることができる，ことを確認した．

今後，この手法を変奏曲以外に適用することを考えている．しかし，本研究では EM アルゴリズムをそのまま使うためには，テーマと変奏曲の類似性がガウス分布的に推定できるという仮定を利用している．しかし，この仮定は変奏曲以外

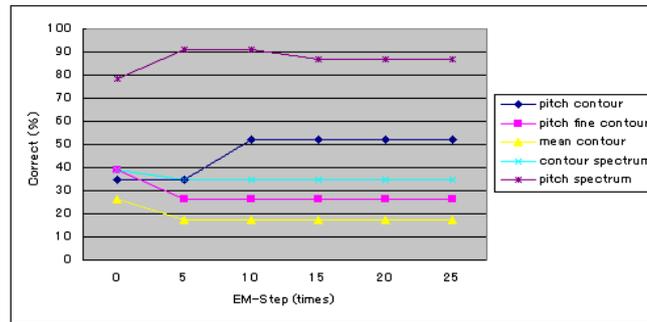


図 2.6: 類似度 50 % 以上

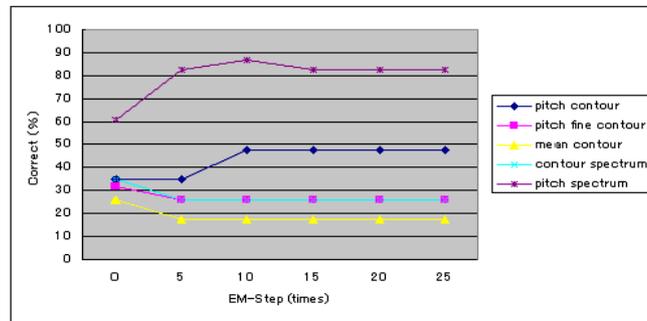


図 2.7: 類似度 60 % 以上

の楽曲で成り立つとは限らず、さらにガウス分布に従うという仮説も検証の必要がある。

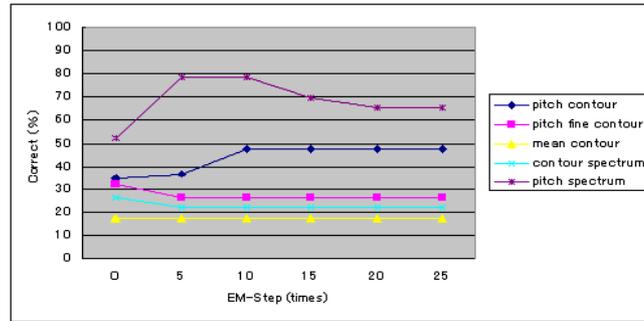


図 2.8: 類似度 66.7 % 以上

	contour(回数)						
	D2	D1	S	U1	U2	U3	U4
K.265 テーマ	4	36	38	14	0	2	6
Op.76 テーマ	9	26	54	14	0	11	4
K.265 v2	0	5	8	1	0	0	1

表 2.6: 特徴量

## 第3章 確率過程に基づく複旋律分類

### 3.1 前書き

本稿では、複旋律で構成された楽曲から特徴値を抽出し、未知の複旋律を分類していずれかラベルを付与する手法(旋律の自動分類)を論じる。これらは、本来手書き文字認識や手書き絵による類似検索と同様に、内容検索を目的とし、曲名や作品番号といったメタ情報を仮定しない。

旋律は主要な印象を与える役割を有しており、この機能を用いた応用は広範囲に想定できる。例えば、音楽作業(作曲・編曲)に対して直接的な支援を行うことや、同じクラスの曲を検索し著作権に抵触しそうな候補の絞込みを行うことができる。実際、音楽における著作権管理では、歌詞と異なって、旋律で判断することが多い<sup>1</sup>。また、インターネットを介して得た音楽コンテンツの自動管理においても旋律による手がかりを与えることができる。

音楽情報処理分野の研究は60年代から開始され、マルチメディア処理技術(信号処理からメタ情報による検索まで)や情報検索理論との関連(Nグラム、ベクトル空間モデル、検索エンジン等)が論じられてきた。

しかし、音楽情報検索において多くの手法は単旋律を元にした手法であるのに対し、広く利用可能であり、また面白い音楽の大部分は複旋律で構成される。複旋律の分類が困難な点は、移調や転調、拍子の変化がパートごとに異なり、複旋律の構成は、その異なった音同士が重なり合い構成されるため、何が主旋律かという要素が不明瞭になる点にある。

他の情報検索分野と比較して、音楽分野での分類問題は特に難しい点がある。まず、音楽情報をモデル化するのにデータモデリング機能が著しく欠けていることが挙げられる。第2に旋律の特定が難しい。実際、旋律の特徴量を定量的に定義することができず、改ざんによる旋律の同定を扱うことさえできない。第3に、旋律の抽出方法が不明であることも重要である。通常、主たる単旋律(テーマ)が中心になるが、編曲の過程で和音、楽器・音色などが付与される。テーマである旨の明示的な記述表現もない。第4に、旋律の類似性の判定(何が答えか?)が明確

<sup>1</sup>歌詞と切り離れた、音楽そのもの「旋律、和音、リズム、形式の4要素が一体となった楽曲」(東京地方裁判所昭和43年5月13日)も独立に著作物となり、保護の対象となる。

に定義できず，例えばクラス所属を確率，ランキング表現など，いずれにすべきか基準が定まらない [18]．最後に，効率よいデータ管理・データ操作に関して，どのような技術が必要かが不明であることも重要である [15]．

このような音楽情報処理の問題は，他の人工知能関連のテーマと同様に，分類問題に帰着させて定式化し，情報検索手法や機械学習手法により，自動計算するのが現実的である．

機械学習手法を用いた帰納推論では，分類クラス(ラベル)が明記された幾つかの訓練データから一般的なパターン(分類器)を抽出し，これを利用して未知データを分類する．正確に分類するには大量の訓練データを必要とするが，ラベルをつける過程は通常人間が行うため，時間がかかり主観的で信頼性にかけたものになりがちである [11]．インターネットを介して得るデジタル音楽のラベルの多くは誤っているか無記名であり，ラベルを正確につける手間は困難な状況になっている．近年これに対応するために，不完全データによる分類器生成手法(ブートストラップ手法)が提案され，とくにEM アルゴリズムが代表的である [27, 23]．

しかし，今までの手法は単旋律分類には効果的であるが，音の構成が不明瞭である複旋律楽曲に直接適用することができない．それに対し，広く利用可能である音楽データは複旋律楽曲である．従って，複旋律楽曲を扱えるような手法が必要となる．本稿では，あらかじめ楽譜データを処理することにより，複旋律楽曲の分類が確率的で計算できるよう，新しい特徴値を提案する．特徴値の抽出には，ベクトル空間に射影した音のスペクトル表示を用いる．

本研究の目的は，複旋律楽曲において確率過程を適用するのに適した特徴量を抽出し，確率過程に基づき旋律を分類することにある．具体的には，複旋律楽曲において小節単位で特徴量を抽出し，旋律特徴量からマルコフモデルを作成し，生成確率を計算することで旋律の分類を行う．

本稿では，音楽に関する基本知識 [26]，および情報検索技術の基本知識を仮定する [5, 29]．

第2章で，準備と必要な定義を述べ，第3章で音楽情報の特徴量を要約し，本研究で扱う特徴量を定義する．第4章では，主として分類に利用するマルコフ過程およびマルコフモデルの適用方法を示し，この特徴や利点を述べる．第5章では実験結果を述べ，本手法の有効性を示す．

## 3.2 旋律の表現

本稿における旋律 (melody) とは，全ての音に対して開始時間や高さ (pitch)，長さ (duration) を表すシンボリックなものである．単旋律 (monophony) とは，同時に発生する音はせいぜい1音しかないものを指す．ホモフォニー (homophony) と

は、同時に発生する音はせいぜい1組の和音しかないものを指し、和音間の音の重なりはないものとする。複旋律 (polyphony) とは、これらの制限がないものとする。

楽曲は複数の音から構成される [26]。ピッチ (pitch) とは音の調子、即ち音の高さを表す。これは周波数で規定され、代表的に記号で表現される。例えば、A (ラ) の音は 440Hz であり、その倍の高さの音を 1 オクターブ (octave) という。例えば 880Hz は A の 1 オクターブ高い a を表す<sup>2</sup>。ピッチを音 (tone) とも呼ぶ。

音程 (interval) はピッチ間の距離であり、1 オクターブ離れた音程を 12 等分した単位を半音 (semitone) という。例えば A から a までは A, A#, B, C, C#, D, D#, E, F, F#, G, G# と表す。G#, A#, C#, D#, F# はそれぞれ Ab, Bb, Db, Eb, Gb と表す。全音階的 (diatonic) な音の並びとは、A, B, C, D, E, F, G, a または C, D, E, F, G, A, B, c のように、半音がそれぞれ 2, 1, 2, 2, 1, 2, 2 または 2, 2, 1, 2, 2, 2, 1 ずつ離れた配置を言う。前者を短調 (minor) 後者を長調 (major) という。また A, C をそれぞれの主音 (key), +7 音を属音 (dominant) という。全音階的でない並びを無調的 (atonal) という。移調 (transposition) とは、主音を短調・長調の並びを保ったまま相対的に移動することをいう。

各ピッチには長さ (duration) がある。和音 (chord) は複数のピッチの並列配置であり、これらは同時に演奏されることを想定している。

五線譜 (five horizontal lines あるいは score) は、音の高さと長さを記入した記号 (音符) を順次的あるいは並列的に記述した表現である。旋律は音符の並びで、和音は音符の重なりで記述される。音符の長さの合計を拍 (beat) という。小節 (bar) とは同じ拍数を持つ (五線譜の) 部分記述であり、拍子 (time signature) によって小節当たりの拍数と拍質を表す。特に、リズム (rhythm) とは拍の規則的な並びをいう。

音楽を表現するためには、多くの表情つけがなされている。これらは大きく 4 つに分けられ、速度・強弱に関するもの、発想に基づくもの、奏法によるもの、装飾音に関するものから成り、五線譜上に直接記載される。しかしこれらを組み合わせても楽曲の記述は容易ではない。例えば、先頭の数小節を与え何らかの類似曲が想定されたとしても、リズムや表情付け、速度等で大きい差があるかもしれない。

楽曲を計算機で表現するには、これまで信号レベル (CD 等でのアナログ的变化を表現)、操作レベル (MIDI など、輝度・時間のゆれを表現) 及び楽譜レベル (表現・解釈の多様性を表現) でなされている。本稿では、特徴量の記述を目的とするため、旋律を五線譜を用いた楽譜レベルの表現を利用する。

<sup>2</sup>以下では 1 オクターブ高いピッチを小文字で表す。

### 3.3 旋律特徴量

旋律を分類するためには、当該音楽の有する意味表現を考慮する必要がある。しかし旋律を分類するために記述される特徴量としてどのようなものが有効なのだろうか？ 旋律を分類するために指定される条件は、不完全である場合が多く、その大半は部分的である。

本研究では、旋律から特徴量を抽出し、ベクタ空間モデル [5, 29] を用いて特徴項目ベクトル空間へ写像する。各次元は特徴項目に対応するため、どの旋律も空間内の点で表せる。旋律質問の解はベクトル空間内の類似した点に対するランク付けで得られる [20]。空間内では伝統的な余弦質問で定義する。即ち  $\cos \theta = \frac{X \cdot Y}{|X| |Y|}$  を計算し、1.0 に近いほど類似していると判断する。ここで  $X, Y$  は空間ベクトルを、 $X \cdot Y$  は内積を、 $|X|$  はベクトル長を表す。

#### 3.3.1 単旋律における特徴量

これまで提案された旋律のための特徴量記述では、旋律輪郭 (Melody Contour) が代表的である [1, 7, 18]。旋律の輪郭情報のうち、ピッチ輪郭 (pitch contour) 表現は多くの研究で検討されている。ピッチ輪郭表現とは、単旋律にだけ有効であり、直前の音と比べてピッチが相対的に”高い”、”低い”、”同じ”という状態をそれぞれ文字 U, D, S で表し、輪郭を文字列として記述する<sup>3</sup>。この表現は、ピッチに対して相対的なので移調に強いが、雑音に反応して文字列が変わりやすく、旋律検索は不完全文字列一致処理の実行を意味する。

しかし、ピッチ輪郭方式は数多くの問題を含む [17]。大規模楽曲の記述から主旋律を抽出することは難しいため、多声音楽から各声を旋律と見なした輪郭を記述し、これらの結合から主旋律を想定する、という組み合わせ的探索を行わざるを得ない [2, 3]。旋律記述の記述に拍子・音符長を含まないため、判別のため多くの情報を必要とする。また、不完全一致検索手続きは多項式時間でよいことが知られるものの複雑で、(全走査を除けば)2 次記憶域上での処理効率を向上させる方法は知られていない。輪郭抽出には雑音除去のための”しきい値”の設定が容易ではなく、生成される輪郭を規定する。

旋律が五線譜による楽譜で記述されるとき、小節単位にピッチスペクトル (pitch spectrum) を導入し特徴量とすることができる [10]。ピッチスペクトルとは、小節内に生じる各音符を長さで集積したヒストグラムを言う。このアイデアは信号処理では提案されているが、楽譜レベルで特徴量化するというアプローチはない [22]。ピッチスペクトルは (ギターコードのような) 小節全体に和音を指定したものと

<sup>3</sup>開始文字は相対位置を定めることができないので - で表す。

異なる．ピッチ輪郭とは異なって，小節単位の特徴を集約しているため，旋律の輪郭を表してはいない．むしろ，小節間の変化を記述しているため，旋律の時系列表現とも異なる．ピッチスペクトルは不完全な旋律に生じる問題を解消することができる．実際，装飾音・旋律・長短調・揺らぎがピッチスペクトルで局所化する．

ピッチスペクトルの問題は 移調に変動しやすいこと，および長調・短調の変化に対応しづらいことであることが知られている [10]．前者については，旋律から主音(キー)を決定しテーマへの移調を行えばよい．しかし後者の問題については対応が難しく，より高度な対応が必要である．

### 3.3.2 複旋律における特徴量

複旋律は，単旋律やホモフォニーのような制限事項がないため，表現される旋律の構成は，無限に近い組み合わせが考えられる．音の構成要素が複雑に関係しているため，単旋律における手法を直接適用することができない．そこで，楽譜をあらかじめ処理することによって対応する方法が考えられる [14]．前処理を行うことで，確率的に計算することが可能となる．

本稿で提案する特徴量は，各小節内で出現する音すべてをベクトルに射影し，音長のスペクトル列で表現することを提案する．この手法による特徴量は，現れる全ての音から構成され，また小節内の音の構成も見ることができるという利点がある．

#### カエルの歌 旋律特徴量の例

カエルの歌を複旋律情報で表現すると次のようになる．

$$\{C:1, D:1, E:1, F:1\}, \{C:2, D:2, E:2, F:1\}, \{C:1, D:1, E:2, F:1, G:1, A:1\}, \{E:2, F:2, G:2, A:1\}, \{C:2, E:1, F:1, G:1\}, \{C:4\}, \{C:3, D:1, E:1, F:1\}, \{C:2, D:2, E:2, F:1\}$$

### 3.3.3 本稿で提案する旋律特徴量

本稿では，旋律分類を行うための特徴量を，各小節内で出現する音を和音とみなす手法を導入し，マルコフモデルによって分類精度を向上できるかを論じる．



図 3.1: カエルの歌五線譜

各小節内で出現する音すべてをベクトルに射影し，音長のスペクトル列で表現する方法を提案したが，さらに，音長上位  $n$  音にしぼり，その  $n$  音を和音としたものを特徴量（特徴量和音）と定義する．小節単位の和音を取ることで，主旋律の進行や特徴を得られると考えられる．ここで構成された特徴量和音とは，音楽理論とは無関係の新しい和音の要素である．

#### カエルの歌 旋律特徴量の例

カエルの歌の先頭 4 小節を，小節内の出現音の長さ上位 3 音を和音とすると次のようになる．

$$1 \text{ 小節目} : \{C : 1, D : 1, E : 1, F : 1\} = \{CDE, CDF, CEF\}$$

$$2 \text{ 小節目} : \{C : 2, D : 2, E : 2, F : 1\} = \{CDE\}$$

$$3 \text{ 小節目} : \{E : 1, F : 1, G : 1, A : 1\} = \{EFG, EFA, FGA\}$$

$$4 \text{ 小節目} : \{E : 1, F : 1, G : 1\} = \{EFG\}$$

### 3.4 マルコフ過程とマルコフモデル

以下では，あらかじめクラスが与えられている旋律  $d$  を用いて，未知旋律をいずれかのクラスに分類する．本研究では，マルコフモデルに基づいた分類手法を用いる． $d$  は，特徴量  $w_j$  を用いた多重集合  $\{w_1, \dots, w_m\}$  と仮定する．

#### 3.4.1 マルコフ過程

旋律  $d = \{w_1, \dots, w_m\}$  に，クラス集合  $C = \{c_1, \dots, c_n\}$  のいずれかの値を割り当てる（分類する）ことを考える．旋律  $d$  の同時確率  $P(d)$  は，音符列の出現する確率であり，条件付確率の積で表現することができる．しかし，全ての音符の組

み合わせに対して条件付確率を  $P(w_j|w_1\dots w_{j-1})$  を推定することは現実的には難しいため、ある事象の確率がその直前の  $N$  個の事象だけに依存すると仮定すると、近似的な式で表すことができる。これを  $N$  重マルコフ過程という。 $N$  グラムモデルにおける生成確率は

$$P(d_n|d_{n-1}) = P(w_j|w_{j-N}) \quad (3.1)$$

となる。つまり、条件付確率  $P(d_n|d_{n-1})$  が求めれば、旋律全体の生成確率の計算が可能となる。しかし、全ての発生する音符列の組み合わせを考えたとき、 $P(d_n|d_{n-1})$  の計算は現実的に困難である。例えば、主な三和音のコード列（長・短・増・減の 48 種類）の生成確率を考えた場合、16 小節で考えたとしても、1 旋律あたり  $48^{16}$  個もの値を推定しなければならない。そこで本稿においては、もっとも単純な  $N = 1$  のときの単純マルコフ過程を考える。単純マルコフ過程であれば、計算量は特徴量数のみに限定され、計算量は少量で済む。

### 3.4.2 マルコフモデルによる文書分類

マルコフモデルとは、現在の状態を限られた過去の状態の積計算で求められる生成確率を導く手順である。一般に、与えられた単語列を  $w = \{w_1, \dots, w_m\}$ ,  $m > 0$  としたとき、求まる生成確率は

$$P(w_1^m) = \prod_{j=1}^m P(w_j|w_1^{j-1}) \quad (3.2)$$

である。本稿では、過去の状態は 1 つ前までの単語列、つまり直前で発生した特徴量しか影響されない単純マルコフモデルを用いる。単純マルコフモデルは以下のように表現できる：

$$P(d) = P(w_j|w_{j-1}) \quad (3.3)$$

単純マルコフモデルによる分類は、まず  $d$  がクラス  $c$  に属する確率  $P(c|d)$  の条件付確率を求めることである。この条件付確率は、生成確率を求めることと同等である。

マルコフモデルを分類操作に適用するアルゴリズムは以下ようになる。

1. 入力：クラスあり旋律集合，未知旋律集合
2. あらかじめクラスの与えられている旋律集合から，マルコフモデル（遷移確率）を生成

3. (2) で生成されたマルコフモデル (遷移確率) を用いて, 未知旋律のクラス所属確率  $P(c_i|d)$  を推定する
4. 出力: 各未知旋律のラベル  $c_k$  は次のように定める:

$$c_k = \text{ArgMax}_{c \in C} P(c|d)$$

マルコフモデルでは, 状態遷移が存在しない確率 (0 確率) が選ばれると解が求まらない.

### 3.4.3 旋律分類への適用

複旋律楽曲において, マルコフモデルを旋律分類に適用する方法を論じる.

以下では  $C = \{c_1, \dots, c_n\}$  はクラス集合を表す. あらかじめクラスを与えられた旋律  $e_c$  はいずれかのクラスに含まれる:  $c \in C$ .  $D$  はすべての未知旋律を表し,  $w_i$  はデータ内の各特徴量を表す. また,  $N(w_i, d_j)$  は旋律  $d_j$  における特徴量  $w_i$  の特徴値である. さらに,  $P(c_k|d_j)$  は前述の旋律  $d_j$  がクラス  $c_k$  に属する確率を表す.

$$P(c|d) = \prod_{i=1}^{|m|} P(w_i|w_{i-1}) \quad (3.4)$$

旋律  $d_j$  が属するクラスは  $c_k$  に属する確率  $P(c_k|d_j)$  が最大となるものを選び, 即ち最尤法に基づくとする.

本稿では, 主題と変奏曲を比較する際, 変奏曲内の和音で主題に現れない和音が存在する可能性が十分にある. これまでの手法を適用した場合, 得られる確率が 0 になる可能性が非常に高い. この問題を解決するために, 通常修正のいずれかの種類が紹介される, しかし, 確率値が積計算によってより小さい値になりがちであるため, 我々が予想しない分類結果になるかもしれない. 従って, 容易に値の修正を決定することができない. かわりに本稿では, それぞれの 1 対の特徴量と和音の間に類似性の考えを導入し, 確率値を調整する.

2 つの特徴量と和音  $w, w'$  という条件のもとで, 我々は類似性  $\text{sim}(w, w')$  を余弦値と定義する:

$$\text{sim}(w, w') = \frac{w \cdot w'}{|w| |w'|} \quad (3.5)$$

そして, 遷移確率  $P(w_i|w_{i-1})$  に  $w_{i-1}$  から  $w_i$  までに起こるすべての可能な和音の上に類似性をかけあわした値の和集合を  $P'(w_i|w_{i-1})$  と定義する.

$$P'(w_i|w_{i-1}) = \sum_g P(w_i|g) \times \text{sim}(g, w_{i-1})$$

この値は、和音  $g$  だけに関する、 $w_i$  を通過し、そして 0 以外の類似性を持っているものであり、マルコフモデルより容易に選ぶことが可能である。また、重みを加えられた生成確率は次で与えられる。余弦値の計算は、特徴を取り出す全ての旋律に対し行う必要がある。

$$P'(c|d) = \prod_{i=1}^{|m|} P'_c(w_i|w_{i-1}) \quad (3.6)$$

ここで重み付けされた確率  $P'(c|d)$  は通常確率値とは異なるが、最尤推定を応用した分類であることから、あらかじめ分類をするすべての音楽について処理し、最尤推定によって所属クラスを決定する。

### マルコフモデル クラス分類例

特徴量表記によるマルコフモデルを用いたクラス分類例を示す。“カエルの歌 (A)”、“キラキラ星変奏曲 (B)” をラベルとし、その旋律が  $d_1, d_2$  として与えられている ( $L_1 = \{d_1, d_2\}$ ) として、“歓喜の歌” ( $d_3$ ) をクラス分類する ( $L_2 = \{d_3\}$ )。ここで利用する特徴量と和音は 3 和音とする。



図 3.2: モーツァルト:キラキラ星変奏曲



図 3.3: ベートーベン:交響曲第 9 番

ABC 表記による先頭 8 小節、もしくは 9 小節の特徴値はそれぞれ次のようになる。

$$d_1: \{CDEF, -\}, \{EDC, CDEF\}, \{EFGA, EDC\}, \{GFE, EFGA\}, \{CC, GFE\}, \\ \{CC, CC\}, \{C/2C/2D/2D/2E/2E/2F/2F/2, CC\}, \{EDC, C/2C/2D/2D/2E/2E/2F/2F/2\}, \\ \{CDE\}$$

$d_2$ : {CCGG,CCEC}, {AAGG,FCEC}, {FFEE,DBCA}, {DD3/4E/4C2,FGC},  
 {CCGG,CCEC}, {AAGG,FCEC}, {FFEE,DBCA}, {DD3/4E/4C2,FGC}  
 $d_3$ : {FFGA}, {AGFE}, {DDEF}, {F3/2E/2E2}, {FFGA}, {AGFE}, {DDEF},  
 {E3/2D/2D2}

小節単位で出現音長をまとめ、その小節における特徴量と和音を決定する。特徴量と和音となり得る候補が複数ある場合は、音の高いものから構成される和音を選択する。また音の進行から、 $d_1$  と  $d_2$  についてマルコフモデルを作成する。

小節	$d_1$	$d_2$	$d_3$
1	{C:1, D:1, E:1, F:1}	{C:5, E:1, G:2}	{F:2, G:1, A:1}
2	{C:2, D:2, E:2, F:1}	{C:2, E:1, F:1, G:2, A:2}	{E:1, F:1, G:1, A:1}
3	{C:1, D:1, E:2, F:1, G:1, A:1}	{C:1, D:1, E:2, F:2, A:1, B:1}	{D:2, E:1, F:1}
4	{E:2, F:2, G:2, A:1}	{C:13/4, D:7/4, E:1/4, F:1, G:1}	{E:5/2, F:3/2}
5	{C:2, E:1, F:1, G:1}	{C:5, E:1, G:2}	{F:2, G:1, A:1}
6	{C:4}	{C:2, E:1, F:1, G:2, A:2}	{E:1, F:1, G:1, A:1}
7	{C:3, D:1, E:1, F:1}	{C:1, D:1, E:2, F:2, A:1, B:1}	{D:2, E:1, F:1}
8	{C:2, D:2, E:2, F:1}	{C:13/4, D:7/4, E:1/4, F:1, G:1}	{D:5/2, E:3/2}
9	{C:1, D:1, E:1}		

表 3.1: 出現和音

マルコフモデルにより、 $d_3$  にクラス割り当てを行う。 $d_3$  のクラス所属確率は上記のそれぞれのマルコフモデルに類似度を考慮した値から求められ、それぞれ次の値となる。

$$P(A|d_3) = 0.0181 \times 0.0181 \times 0.0375 = 1.23 \times 10^{-5}$$

$$P(B|d_3) = 0.0417 \times 0.0347 \times 0.0170 = 2.46 \times 10^{-5}$$

小節	1	2	3	4	5	6	7	8	9
$d_1$	DEF	CDE	EGA	EFG	CFG	C	CEF	CDE	CDE
$d_2$	CEG	CGA	EFB	CDG	CEG	CGA	EFB	CDG	
$d_3$	FGA	FGA	DEF	EF	FGA	FGA	DEF	EF	

表 3.2: 特徴量和音

	CDE	CEF	CFG	DEF	EFG	EGA	C
CDE	0.5	0	0	0	0	0.5	0
CEF	1	0	0	0	0	0	0
CFG	0	0	0	0	0	0	1
DEF	1	0	0	0	0	0	0
EFG	0	0	1	0	0	0	0
EGA	0	0	0	0	1	0	0
C	0	1	0	0	0	0	0

表 3.3: マルコフモデルの遷移確率：カエルの歌

この結果,  $d_3$  が最も近いクラスは  $B$  であり, ”歓喜の歌” は ”キラキラ星” に割り当てられた.

なお, 本稿で用いられる旋律内での各特徴量は, その定義に従って与えられる.

## 3.5 実験

以下の実験では幾つかの変奏曲を扱う. 変奏曲は1つのテーマと複数の変奏曲から構成されるため, ラベル付き訓練データはテーマの数だけあると考える. これ以外はラベル無しデータとみなし, 旋律分類の対象とする.

### 3.5.1 準備

本稿では, 複旋律で構成されたピアノ変奏曲を用いる. 使用する楽曲は, Mozart キラキラ星変奏曲 K.265, Schubert 即興曲 Op.142-3, および Beethoven トルコ行進曲による6つの変奏曲 Op.76 の3つの変奏曲であり, それぞれ主題(テーマ)と12曲, 5曲, 6曲の変奏(バリエーション)からなる. これらは MIDI データとして表現されているものを ABC フォーマット [21] に変換して実験に使用する.

	CDG	CEG	CGA	EFB
CDG	0	1	0	0
CEG	0	0	1	0
CGA	0	0	0	1
EFB	1	0	0	0

表 3.4: マルコフモデルの遷移確率：キラキラ星

最初に，各楽曲をラベル付旋律集合とラベルなし旋律集合に分ける．ラベル付旋律集合は変奏曲テーマ（主旋律）3曲からなるとし，残り 23 の変奏曲はラベルなし旋律集合を構成する．各楽曲を表す ABC ファイルからラベル付旋律集合は全小節分の特徴量，ラベルなし旋律集合は先頭 4 小節分の特徴量，または先頭 8 小節分の特徴量を取り出す．

本実験で抽出する特徴量は，小節内の出現音の長さ上位 3, 4, 5 音を，その小節の特徴量和音とする．

### 3.5.2 評価方法

本実験では 3 つのクラス "Mozart", "Schubert", "Beethoven" を設定し，マルコフモデルを用いた分類器は全 23 変奏曲をいずれかのクラスに分類する．変奏曲が "正しく分類" できたとはそのテーマがラベルの作曲者によって作曲されたときとする．形式的に正解率を次のように定義する：

$$\frac{\text{正しく分類された変奏曲数}}{\text{総変奏曲数}}$$

本稿における総変奏曲数は 23 となる．

### 3.5.3 実験結果

変奏曲テーマ（主旋律）から作成したマルコフモデルを図 3.4 から図 3.9 に示す．また，特徴量を先頭 4 小節を用いた際と先頭 8 小節を用いた際のマルコフモデルで計算された生成確率を表 3.5，表 3.6，図 3.13 から図 3.16 に，各テーマに対する正解数，総正解数，正解率を表 3.7 に示す．

全ての楽曲において，4 小節より 8 小節の特徴量を用いた場合に，同等あるいは高い正解率を示した．特に 3 和音のときに "Schubert" の正解率が 60 % も向上した．また，4 小節においても "Mozart" は高い正解率を得た．

	CDG	CEG	CGA	DEG	EFG	EFB
CDG	0	1	0	0	0	0
CEG	0	0.25	0.5	0	0.25	0
CGA	0	0	0	0	0	1
DEG	0	0	0	0	1	0
EFG	0	0.5	0	0.5	0	0
EFB	1	0	0	0	0	0

図 3.4: マルコフモデル : Mozart 3 和音

	CFG	CFA	CFB	DFB	DGB	DAB	EGB	FGA
CFG	0	0	0	1	0	0	0	0
CFA	0	0	0	0	1	0	0	0
CFB	0	0	0	1	0	0	0	0
DFB	0	0.444	0.111	0.111	0.111	0	0.222	0
DGB	0	0	0.333	0	0	0.333	0.333	0
DAB	0	0	0	0.5	0.5	0	0	0
EGB	0	0	0.5	0	0	0	0	0.5
FGA	1	0	0	0	0	0	0	0

図 3.5: マルコフモデル : Schubert 3 和音

### 3.5.4 考察

結果から明らかのように，特徴量の小節数を増やすことで結果を向上させることができる．実際，4 小節と少ない小節数で 73.9 % の正解率だったのに対し，8 小節においては 91.3 % と 17.4 % の向上が見られた．この理由として，特徴値の取り方が考えられる．変奏曲が他の主題と類似していることから間違っただ分類が行われた．具体的には，”Schubert” テーマの第 3 変奏曲は，{CFB}{CEF}{CEF,CFB}{CFB}{CFB}{CEF}{CEB}{CFB} という和音進行をとる．この和音進行で出現する音は C, E, F, B のみである．先頭 4 小節では，”Mozart” テーマの方が ”Schubert” テーマと同じ和音が多く現れたため，正解との判別が難しかったと考えられる．8 小節に拡張した場合，”Mozart” テーマには ”Schubert” のテーマとは類似度が低い和音が現れ，生成確率が大きく減少したことで正解へと向上したと考えられる．なお，表 3.8 における下線部は一致する和音を指す．

正しく分類されなかった変奏曲の特徴として，いくつかの原因が挙げられる．1 つ

	CDA	CEA	CAB	DFA	DFB	DGA	EGA	DF	DB
CDA	0	0	0	0	0	0	0	1	0
CEA	0	0	0	0	0	0	1	0	0
CAB	0	0	0	1	0	0	0	0	0
DFA	0.125	0	0	0.625	0.125	0.125	0	0	0
DFB	0	0.5	0.25	0	0	0	0	0	0.25
DGA	0	0	0	1	0	0	0	0	0
EGA	0	0	0	1	0	0	0	0	0
DF	0	0	0	0	1	0	0	0	0
DB	0	0	0	0	1	0	0	0	0

図 3.6: マルコフモデル : Beethoven 3 和音

	CDEG	CDFG	CFG A	DEFG	DEGB	EFAB	CEG
CDEG	0	0	0	0	0	0	1
CDFG	0	0	0	0.667	0	0	0.333
CFG A	0	0.2	0	0	0	0.8	0
DEFG	0.5	0	0	0	0.5	0	0
DEGB	0	0	0	1	0	0	0
EFAB	0	1	0	0	0	0	0
CEG	0	0	1	0	0	0	0

図 3.7: マルコフモデル : Mozart 4 和音

	CDFA	CDFB	CEFA	CEFB	CEGB	CFG A	CFGB	DEGB	DFGA	DFGB	DFAB	DGAB	DFB
CDFA	0	0	0	0	0	0	0	1	0	0	0	0	0
CDFB	0	0	0	0	0	0	0	0	0	0	0	0	0
CEFA	0	0	0	0	0	0	0	1	0	0	0	0	0
CEFB	0	0	0	0	0	0	0	0	0	0	0	0	1
CEGB	0	0	0	0	0	0	0.5	0	0.5	0	0	0	0
CFG A	0	0	0	0	0	0	0	0	0	1	0	0	0
CFGB	0	0	0	0	0	0	0	0	0	0	1	0	0
DEGB	0	0	0	0.5	0	0	0	0	0	0	0.5	0	0
DFGA	0	0	0	0	0	1	0	0	0	0	0	0	0
DFGB	0	0	0	0	1	0	0	0	0	0	0	0	0
DFAB	0	0.2	0	0	0	0	0	0	0	0	0.2	0.4	0.2
DGAB	0	0	0	0	1	0	0	0	0	0	0	0	0
DFB	0.5	0	0.5	0	0	0	0	0	0	0	0	0	0

図 3.8: マルコフモデル : Schubert 4 和音

は、変調によるものである。4 小節、8 小節において間違いとなっている ”Mozart” テーマの第 12 変奏曲は、主題が  $\frac{4}{4}$  拍子に対し、変奏曲は  $\frac{12}{8}$  拍子となっている。調子が変わることで、小節単位による特長量の取り方では、小節をまたがった場合に音のずれをカバーできていないことが分かる。

このことから 4 小節という少ない小節数の場合、特徴的な差を得られるような特徴値の出現パターンが現れていないことから、本稿で提案する特徴量では正解率向上の限界があると思われる。しかし、本実験において 4 小節でも 78.3 % の正解率が得られていることから、本手法は複旋律分類において非常に有効であると考えられる。

また逆に、特徴量と和音の大きさを拡大していくにつれて、正解率が低下していることが分かる。そして、正解の旋律が ”Mozart” クラスに集中している。これは、構成される音が多くなるにつれて、特徴的な情報までが失われてしまってい

	CDFB	CDGA	CEGA	CEAB	DEGA	DFGA	DFA	DFB	DF	DB
CDFB	0	0	0	0.5	0	0	0	0	0	0.5
CDGA	0	0	0	0	0	0	0	0	1	0
CEGA	0	0	0.5	0	0	0	0.5	0	0	0
CEAB	0	0	0	0	0	0	1	0	0	0
DEGA	0	0	0	0	0	1	0	0	0	0
DFGA	0	0	0	0	0	0	1	0	0	0
DFA	0.143	0.143	0	0	0.143	0	0.571	0	0	0
DFB	0	0	1	0	0	0	0	0	0	0
DF	0	0	0	0	0	0	0	1	0	0
DB	0	0	0	0	0	0	0	1	0	0

図 3.9: マルコフモデル : Beethoven 4 和音

	CDEFG	CDEGB	CEFGA	DEFAB	DEFG	CEG
CDEFG	0	0	0	0	0.667	0.333
CDEGB	0	0	0	0	0.5	0.5
CEFGA	0	0	0	1	0	0
DEFAB	1	0	0	0	0	0
DEFG	0	1	0	0	0	0
CEG	0	0	1	0	0	0

図 3.10: マルコフモデル : Mozart 5 和音

	CDEFA	CDEGB	CDFGA	CDFAB	CEFGA	CEFGB	CEGAB	DEFGB	DFGAB	DFGB	DFB
CDEFA	0	0	0	0	0	0	0	1	0	0	0
CDEGB	0	0	1	0	0	0	0	0	0	0	0
CDFGA	0	0	0	0	1	0	0	0	0	0	0
CDFAB	0	0	0	0.4	0	0	0	0	0.4	0	0.2
CEFGA	0	0	0	0	0	0	0	0	0	1	0
CEFGB	0	0	0	0.5	0	0	0	0	0	0	0.5
CEGAB	0	0	0	0	0	1	0	0	0	0	0
DEFGB	0	0	0	0.5	0	0.5	0	0	0	0	0
DFGAB	0	1	0	0	0	0	0	0	0	0	0
DFGB	0	0	0	0	0	0	1	0	0	0	0
DFB	1	0	0	0	0	0	0	0	0	0	0

図 3.11: マルコフモデル : Schubert 5 和音

ると考えられる。実際， $\{C:2, D:2, E:2, F:1, G:1\}$  と  $\{C:1, D:1, E:2, F:2, G:2\}$  の和音では，上位 3 音ならば  $\{CDE\}$  と  $\{EFG\}$  となり，それぞれは違う構成であると分かるが，上位 5 音ならばいずれの特徴量と和音も  $\{CDEFG\}$  となり，異なる旋律同士がおなじラベルを与えられることになる。その結果として，本実験では類似した旋律はすべて "Mozart" とみなされてしまい，それ以外の曲の正解率があがらなかったと考えられる。これより，特徴量と和音は 3 和音程度で構成することがよいと考えられる。

正解率の比較として，単旋律曲の分類において最も正解率の高かった特徴量を先頭 4 小節でピッチスペクトラムを用い，分類手法に単純ベイズ，および EM アルゴリズムを用いた場合の正解率 [23] を表 3.9 に示す。

単純ベイズ法 (EM loop 0 回) で最高 87 %，EM アルゴリズムで最高 91.3 %を示したが，本研究の手法においても，単旋律から複旋律へと複雑化しているにもかかわらず特徴量抽出を 8 小節用いた場合の正解率が 91.3 %と，単旋律での分類

	CDEGA	CDEAB	CDFAB	DEFGA	DEGAB	CEGA	DFA	DFB	DF	DB
CDEGA	0	0	0	0	0	0	0	0	1	0
CDEAB	0	0	0	0	0	0	1	0	0	0
CDFAB	0	0.5	0	0	0	0	0	0	0	0.5
DEFGA	0	0	0	0	0	0	1	0	0	0
DEGAB	0	0	0	1	0	0	0	0	0	0
CEGA	0	0	0	0	0	0.5	0.5	0	0	0
DFA	0.143	0	0.143	0	0.143	0	0.571	0	0	0
DFB	0	0	0	0	0	1	0	0	0	0
DF	0	0	0	0	0	0	0	1	0	0
DB	0	0	0	0	0	0	0	0	1	0

図 3.12: マルコフモデル : Beethoven 5 和音

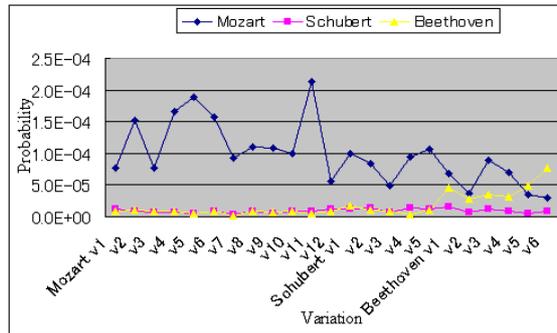


図 3.13: マルコフモデルによる生成確率: 4 和音, 4 小節

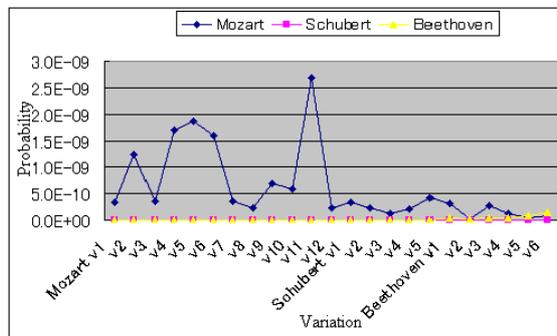


図 3.14: マルコフモデルによる生成確率: 4 和音, 8 小節

と比較しても互角である。本研究の手法は、複旋律分類において有効な抽出方法といえよう。

### 3.6 結び

本研究では、あらかじめクラスが与えられた少量の複旋律楽曲から多く存在する未知旋律を分類する方法を提案した。具体的には、複旋律情報の楽曲を小節単位の音情報を和音とした特徴量も用いることと、マルコフモデルに基づいて計算することにより、分類結果で高い正解率が得られた。この結果、複旋律情報においても高い分類精度を得ることを確認した。

本研究では単純マルコフモデルを用いた推定を行っているが、これは変奏曲分類の楽曲で成り立つかの検証の必要がある。

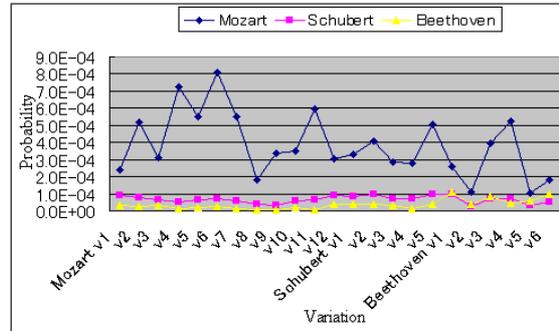


図 3.15: マルコフモデルによる生成確率: 5 和音, 4 小節

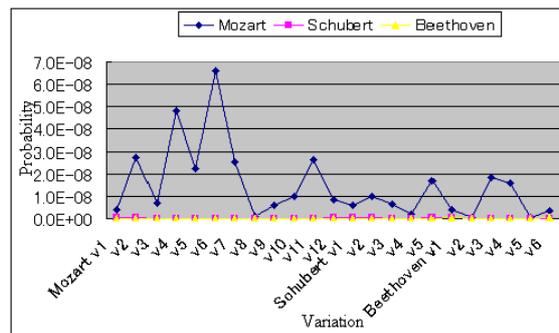


図 3.16: マルコフモデルによる生成確率: 5 和音, 8 小節

	Mozart	Schubert	Beethoven
Mozart v1	$7.497 \times 10^{-5}$	$1.292 \times 10^{-6}$	$6.384 \times 10^{-7}$
v2	$4.895 \times 10^{-5}$	$3.066 \times 10^{-6}$	$2.140 \times 10^{-6}$
v3	$5.604 \times 10^{-5}$	$2.910 \times 10^{-6}$	$3.034 \times 10^{-6}$
v4	$2.257 \times 10^{-4}$	$1.923 \times 10^{-6}$	$9.237 \times 10^{-7}$
v5	$8.335 \times 10^{-5}$	$1.898 \times 10^{-6}$	$1.530 \times 10^{-6}$
v6	$1.827 \times 10^{-4}$	$2.331 \times 10^{-6}$	$7.621 \times 10^{-7}$
v7	$6.125 \times 10^{-5}$	$2.970 \times 10^{-6}$	$9.249 \times 10^{-7}$
v8	$6.417 \times 10^{-5}$	$1.782 \times 10^{-5}$	$1.057 \times 10^{-6}$
v9	$4.434 \times 10^{-5}$	$1.229 \times 10^{-5}$	$9.714 \times 10^{-7}$
v10	$3.936 \times 10^{-5}$	$8.525 \times 10^{-6}$	$1.003 \times 10^{-6}$
v11	$6.801 \times 10^{-5}$	$9.203 \times 10^{-6}$	$1.814 \times 10^{-6}$
v12	$2.415 \times 10^{-5}$	$8.412 \times 10^{-6}$	$1.069 \times 10^{-6}$
Schubert v1	$1.961 \times 10^{-5}$	$1.608 \times 10^{-5}$	$1.155 \times 10^{-5}$
v2	$6.319 \times 10^{-6}$	$3.563 \times 10^{-5}$	$8.079 \times 10^{-6}$
v3	$1.548 \times 10^{-5}$	$2.418 \times 10^{-5}$	$2.429 \times 10^{-6}$
v4	$6.221 \times 10^{-5}$	$1.854 \times 10^{-5}$	$4.496 \times 10^{-7}$
v5	$3.212 \times 10^{-5}$	$2.091 \times 10^{-5}$	$2.068 \times 10^{-6}$
Beethoven v1	$5.943 \times 10^{-6}$	$9.357 \times 10^{-6}$	$6.138 \times 10^{-5}$
v2	$7.563 \times 10^{-6}$	$1.380 \times 10^{-5}$	$6.987 \times 10^{-6}$
v3	$4.065 \times 10^{-5}$	$1.003 \times 10^{-5}$	$2.370 \times 10^{-5}$
v4	$5.372 \times 10^{-5}$	$2.674 \times 10^{-6}$	$7.352 \times 10^{-6}$
v5	$6.174 \times 10^{-6}$	$6.447 \times 10^{-6}$	$2.851 \times 10^{-5}$
v6	$2.026 \times 10^{-6}$	$1.878 \times 10^{-5}$	$5.476 \times 10^{-5}$

表 3.5: マルコフモデルによる生成確率: 3 和音, 4 小節

	Mozart	Schubert	Beethoven
Mozart v1	$2.516 \times 10^{-10}$	$3.959 \times 10^{-14}$	$4.648 \times 10^{-15}$
v2	$9.985 \times 10^{-11}$	$2.385 \times 10^{-13}$	$5.345 \times 10^{-14}$
v3	$2.036 \times 10^{-10}$	$1.764 \times 10^{-13}$	$9.124 \times 10^{-14}$
v4	$4.047 \times 10^{-9}$	$3.852 \times 10^{-14}$	$4.682 \times 10^{-15}$
v5	$2.788 \times 10^{-10}$	$6.669 \times 10^{-14}$	$3.353 \times 10^{-14}$
v6	$2.163 \times 10^{-9}$	$1.132 \times 10^{-13}$	$5.755 \times 10^{-15}$
v7	$1.560 \times 10^{-10}$	$4.688 \times 10^{-14}$	$8.624 \times 10^{-15}$
v8	$7.913 \times 10^{-11}$	$9.745 \times 10^{-12}$	$1.189 \times 10^{-14}$
v9	$1.153 \times 10^{-10}$	$3.767 \times 10^{-12}$	$1.054 \times 10^{-14}$
v10	$1.231 \times 10^{-10}$	$7.571 \times 10^{-13}$	$5.520 \times 10^{-15}$
v11	$2.391 \times 10^{-10}$	$2.059 \times 10^{-12}$	$5.617 \times 10^{-14}$
v12	$2.121 \times 10^{-11}$	$2.293 \times 10^{-12}$	$7.002 \times 10^{-14}$
Schubert v1	$2.394 \times 10^{-12}$	$1.285 \times 10^{-11}$	$5.642 \times 10^{-13}$
v2	$3.857 \times 10^{-13}$	$4.468 \times 10^{-11}$	$2.513 \times 10^{-12}$
v3	$8.988 \times 10^{-12}$	$1.010 \times 10^{-11}$	$5.125 \times 10^{-14}$
v4	$1.977 \times 10^{-11}$	$2.134 \times 10^{-11}$	$2.417 \times 10^{-14}$
v5	$6.763 \times 10^{-12}$	$1.354 \times 10^{-11}$	$7.304 \times 10^{-14}$
Beethoven v1	$5.455 \times 10^{-13}$	$1.606 \times 10^{-12}$	$3.163 \times 10^{-11}$
v2	$7.135 \times 10^{-13}$	$4.240 \times 10^{-12}$	$1.785 \times 10^{-12}$
v3	$5.780 \times 10^{-11}$	$2.013 \times 10^{-12}$	$2.207 \times 10^{-12}$
v4	$3.143 \times 10^{-12}$	$2.966 \times 10^{-13}$	$5.968 \times 10^{-12}$
v5	$8.236 \times 10^{-13}$	$1.124 \times 10^{-12}$	$2.356 \times 10^{-11}$
v6	$2.051 \times 10^{-13}$	$1.486 \times 10^{-11}$	$1.082 \times 10^{-10}$

表 3.6: マルコフモデルによる生成確率: 3和音, 8小節

( )内は和音数	正解数				正解率
	Mozart	Schubert	Beethoven	合計	
(3) 4小節	12/12	2/5	3/6	17/23	73.9 %
(3) 8小節	12/12	5/5	4/6	21/23	91.3 %
(4) 4小節	12/12	0/5	2/6	14/23	60.9 %
(4) 8小節	12/12	0/5	2/6	14/23	60.9 %
(5) 4小節	12/12	0/5	0/6	12/23	52.2 %
(5) 8小節	12/12	0/5	0/6	12/23	52.2 %

表 3.7: 正解率と正解数

	4小節	8小節
Schubert v3	{CFB} {CEF}	+{CEB}
Mozart テーマ	{DFB} {CFA} {DEB} {DGB} {CEF} {CEB} {CFB}	+{DFB}{DAB}
Schubert テーマ	{CEG} {CGA} {CEF} {DEF} {EFA} {EFB} {CDF} {CDG}	追加なし

表 3.8: 出現和音一覧

		EM loop					
		0	5	10	15	20	25
類似度	30	87	91.3	91.3	91.3	91.3	91.3
(%)	50	78.3	91.3	91.3	87.0	87.0	87.0

表 3.9: 正解率：単旋律，ピッチスペクトル

## 第4章 楽曲進行の推定による複旋律分類

### 4.1 前書き

本稿では，複旋律で構成された楽曲から特徴値を抽出し，未知の複旋律を分類していずれかラベルを付与する手法（旋律の自動分類）を論じる．これらは，本来手書き文字認識や手書き絵による類似検索と同様に，内容検索を目的とし，曲名や作品番号といったメタ情報を仮定しない．

旋律は主要な印象を与える役割を有しており，この機能を用いた応用は広範囲に想定できる．例えば，音楽作業（作曲・編曲）に対して直接的な支援を行うことや，同じクラスの曲を検索し著作権に抵触しそうな候補の絞込みを行うことができる．実際，音楽における著作権管理では，歌詞と異なって，旋律で判断することが多い．また，インターネットを介して得た音楽コンテンツの自動管理においても旋律による手がかりを与えることができる．

本研究の目的は，複旋律楽曲において確率過程を適用するのに適した特徴量を抽出し，確率過程に基づき旋律を分類することにある．具体的には，複旋律楽曲において小節単位で特徴量を抽出し，隠れマルコフモデルに基づき，生成確率を計算することで旋律の分類を行う．また同時に，未知旋律に対してコードの推定も行う．

本稿では，音楽に関する基本知識 [26]，および情報検索技術の基本知識を仮定する [5, 29]．

第2章で，準備と必要な定義を述べ，第3章で音楽情報の特徴量を要約し，本研究で扱う特徴量を定義する．第4章では，主として分類に利用するマルコフ過程および隠れマルコフモデルの適用方法を示し，この特徴や利点を述べる．第5章では実験結果を述べ，本手法の有効性を示す．

## 4.2 旋律の表現

本稿における旋律 (melody) とは、全ての音に対して開始時間や高さ (pitch)、長さ (duration) を表すシンボリックなものである。単旋律 (monophony) とは、同時に発生する音はせいぜい 1 音しかないものを指す。ホモフォニー (homophony) とは、同時に発生する音はせいぜい 1 組の和音しかないものを指し、和音間の音の重なりはないものとする。複旋律 (polyphony) とは、これらの制限がないものとする。コード (Chord, 和音) とは、3 音以上の音の積み重ね、またその響きを指す。全てのコードには基準となる音があり、これをルート (Root) という。

楽曲を計算機で表現するには、これまで信号レベル (CD 等でのアナログ的变化を表現)、操作レベル (MIDI など、輝度・時間のゆれを表現) 及び楽譜レベル (表現・解釈の多様性を表現) でなされている。本稿では、特徴量の記述を目的とするため、旋律を五線譜を用いた楽譜レベルの表現を利用する。

## 4.3 旋律特徴量

旋律を分類するためには、当該音楽の有する意味表現を考慮する必要がある。しかし旋律を分類するために記述される特徴量としてどのようなものが有効なのだろうか？ 旋律を分類するために指定される条件は、不完全である場合が多く、その大半は部分的である。

本研究では、旋律から特徴量を抽出し、ベクタ空間モデル [5, 29] を用いて特徴項目ベクトル空間へ写像する。各次元は特徴項目に対応するため、どの旋律も空間内の点で表せる。旋律質問の解はベクトル空間内の類似した点に対するランク付けで得られる [20]。空間内では伝統的な余弦質問で定義する。即ち  $\cos \theta = \frac{X \cdot Y}{|X| |Y|}$  を計算し、1.0 に近いほど類似していると判断する。ここで  $X, Y$  は空間ベクトルを、 $X \cdot Y$  は内積を、 $|X|$  はベクトル長を表す。

### 4.3.1 単旋律における特徴量

これまで提案された旋律のための特徴量記述では、旋律輪郭 (Melody Contour) が代表的である [1, 7, 18]。旋律の輪郭情報のうち、ピッチ輪郭 (pitch contour) 表現は多くの研究で検討されている。ピッチ輪郭表現とは、単旋律にだけ有効であり、直前の音と比べてピッチが相対的に”高い”、”低い”、”同じ”という状態をそれぞれ文字 U, D, S で表し、輪郭を文字列として記述する<sup>1</sup>。この表現は、ピッチに対

<sup>1</sup>開始文字は相対位置を定めることができないので - で表す。

して相対的なので移調に強いが、雑音に反応して文字列が変わりやすく、旋律検索は不完全文字列一致処理の実行を意味する。

しかし、ピッチ輪郭方式は数多くの問題を含む [17]。大規模楽曲の記述から主旋律を抽出することは難しいため、多声音楽から各声を旋律と見なした輪郭を記述し、これらの結合から主旋律を想定する、という組み合わせ的探索を行わざるを得ない [2, 3]。旋律記述の記述に拍子・音符長を含まないため、判別のため多くの情報を必要とする。また、不完全一致検索手続きは多項式時間でよいことが知られるものの複雑で、(全走査を除けば)2次記憶域上での処理効率を向上させる方法は知られていない。輪郭抽出には雑音除去のための”しきい値”の設定が容易ではなく、生成される輪郭を規定する。

旋律が五線譜による楽譜で記述されるとき、小節単位にピッチスペクトル (pitch spectrum) を導入し特徴量とすることができる [10]。ピッチスペクトルとは、小節内に生じる各音符を長さで集積したヒストグラムを言う。このアイデアは信号処理では提案されているが、楽譜レベルで特徴量化するというアプローチはない [22]。ピッチスペクトルは(ギターコードのような)小節全体に和音を指定したものとは異なる。ピッチ輪郭とは異なって、小節単位の特徴を集約しているため、旋律の輪郭を表してはいない。むしろ、小節間の変化を記述しているため、旋律の時系列表現とも異なる。ピッチスペクトルは不完全な旋律に生じる問題を解消することができる。実際、装飾音・旋律・長短調・揺らぎがピッチスペクトルで局所化する。

ピッチスペクトルの問題は 移調に変動しやすいこと、および長調・短調の変化に対応しづらいことであることが知られている [10]。前者については、旋律から主音(キー)を決定しテーマへの移調を行えばよい。しかし後者の問題については対応が難しく、より高度な対応が必要である。

### 4.3.2 複旋律における特徴量

複旋律は、単旋律やホモフォニーのような制限事項がないため、表現される旋律の構成は、無限に近い組み合わせが考えられる。音の構成要素が複雑に関係しているため、単旋律における手法を直接適用することができない。そこで、楽譜をあらかじめ処理することによって対応する方法が考えられる [14]。前処理を行うことで、確率的に計算することが可能となる。

本稿で提案する特徴量は、各小節内で出現する音すべてをベクトルに射影し、音長のスペクトル列で表現することを提案する。この手法による特徴量は、現れる全ての音から構成され、また小節内の音の構成も見ることができるという利点がある。

## カエルの歌 旋律特徴量の例

カエルの歌を複旋律情報で表現すると次のようになる。

$$\{C:1, D:1, E:1, F:1\}, \{C:2, D:2, E:2, F:1\}, \{C:1, D:1, E:2, F:1, G:1, A:1\}, \{E:2, F:2, G:2, A:1\}, \{C:2, E:1, F:1, G:1\}, \{C:4\}, \{C:3, D:1, E:1, F:1\}, \{C:2, D:2, E:2, F:1\}$$


図 4.1: カエルの歌五線譜

## 4.3.3 本稿で提案する旋律特徴量

本稿では、旋律分類を行うための特徴量を、各小節内で出現する音から構成する手法を導入し、マルコフモデルによって分類精度を向上できるかを論じる。

各小節内で出現する音すべてをベクトルに射影し、音長のスペクトル列で表現する方法を提案したが、さらに、音長上位  $n$  音にしぼり、その  $n$  音で構成された集合を特徴量（小節構成音）と定義する。小節単位の音情報を利用することで、主旋律の進行や特徴を得られると考えられる。ここで構成された小節構成音とは、音楽理論とは無関係の新しい和音の要素である。また、小節構成音となり得る候補が複数ある場合は、音の高いものから構成される音の組み合わせを選択する。

## カエルの歌 旋律特徴量の例

カエルの歌の先頭 4 小節を、小節内の出現音の長さ上位 3 音を小節構成音とすると次のようになる。

1 小節目： $\{C:1, D:1, E:1, F:1\} = \{DEF\}$

2 小節目： $\{C:2, D:2, E:2, F:1\} = \{CDE\}$

3 小節目： $\{E:1, F:1, G:1, A:1\} = \{FGA\}$

4 小節目： $\{E:1, F:1, G:1\} = \{EFG\}$

## 4.4 隠れマルコフモデル

以下では、あらかじめクラスが与えられている旋律  $d$  を用いて、未知旋律をいずれかのクラスに分類する。本研究では、隠れマルコフモデルに基づいた分類手法を用いる。  $d$  は、特徴量  $w_j$  を用いた多重集合  $\{w_1, \dots, w_m\}$  と仮定する。

### 4.4.1 隠れマルコフモデル

隠れマルコフモデルとは、実際に観測された出力記号系列から、その出力記号系列が生成されるための状態遷移系列モデルを推測・生成する手順である。一般に、与えられた記号系列からモデルのパラメータを決定することになる。本稿では、状態遷移系列をあらかじめ付与し Forward アルゴリズム (前向きアルゴリズム)、Viterbi アルゴリズムを用いてパラメータを推定する。Forward アルゴリズムとは、以下のようなステップで確率を再帰的に計算するアルゴリズムである。

1. 各状態  $i=1, \dots, N$  に対して前向き確率を初期化： $\alpha_1(i) = \pi_i b_i(o_1)$
2. 各時刻  $t=1, \dots, T-1$ , 各状態  $j=1, \dots, N$  について、前向き確率を再帰的に計算：  

$$\alpha_{t+1}(j) = [\sum_{i=1}^N \alpha_t(i) a_{ij}] b_j(O_{t+1})$$
3. すべての累積確率を合計： $P(o_1^T | M) = \sum_{i=1}^N \alpha_T(i)$

また Viterbi アルゴリズムは、以下のようなステップで求めるアルゴリズムである。

1. 各状態  $i=1, \dots, N$  に対して前向き確率を初期化： $\delta_1(i) = \pi_i b_i(o_1)$
2. 各時刻  $t=1, \dots, T-1$ , 各状態  $j=1, \dots, N$  について、再帰的に計算：  

$$\delta_{t+1}(j) = \max_i [\delta_t(i) a_{ij}] b_j(O_{t+1})$$
3. 最終的な累積確率を計算： $\hat{P} = \max_i \delta_T(i)$

Forward アルゴリズムと Viterbi アルゴリズムはいずれも再帰的に確率を求める計算方法であるが、大きく異なる点は Viterbi アルゴリズムでは状態遷移に対して最大の確率値を用いていることにある。

#### 4.4.2 旋律分類への適用

本稿で扱う分類操作は，平均律コードを状態列として割当ててことでモデルの生成を行う．そして生成されたモデルを用いて，クラス所属確率を推定する．隠れマルコフモデルを分類操作に適用するアルゴリズムは以下ようになる．

1. 入力：クラスあり旋律集合，未知旋律集合
2. あらかじめクラスの与えられている旋律集合から，状態列となる平均律コードを付与しモデルを推定・生成
3. (2) で生成されたモデル（遷移確率）を用いて，未知旋律のクラス所属確率  $P(c_i|d)$  を推定する
4. 出力：各未知旋律のラベル  $c_k$  は次のように定める：

$$c_k = \text{ArgMax}_{c \in C} P(c|d)$$

本稿では，主題と変奏曲を比較する際，変奏曲内の小節構成音で主題に現れない小節構成音が存在する可能性が十分にある．これまでの手法を適用した場合，得られる確率が 0 になる可能性が非常に高い．この問題を解決するために，通常修正のいずれかの種類が紹介される，しかし，確率値が積計算によってより小さい値になりがちであるため，我々が予想しない分類結果になるかもしれない．従って，容易に値の修正を決定することができない．かわりに本稿では，それぞれの 1 対の小節構成音の間に類似性の考えを導入し，確率値を調整する．

2 つの小節構成音  $w, w'$  という条件のもとで，我々は類似性  $\text{sim}(w, w')$  を余弦値と定義する：

$$\text{sim}(w, w') = \frac{w \cdot w'}{|w| |w'|} \quad (4.1)$$

そして，遷移確率  $P(w_i|w_{i-1})$  に  $w_{i-1}$  から  $w_i$  までに起こるすべての可能な小節構成音の上に類似性を掛け合わせた値の和集合を  $P'(w_i|w_{i-1})$  と定義する．

$$P'(w_i|w_{i-1}) = \sum_g P(w_i|g) \times \text{sim}(g, w_{i-1})$$

この値は，ひとつの状態遷移の確率値と同等である．また，重みを加えられた生成確率は次で与えられる．余弦値の計算は，特徴を取り出す全ての旋律に対し行う必要がある．そして，新しい基準は最尤推定のもとで調整される．

$$P'(c|d) = \prod_{i=1}^{|m|} P'_c(w_i|w_{i-1}) \quad (4.2)$$



$d_1$ : {CDEF, -}, {EDC, CDEF}, {EFGA, EDC}, {GFE, EFGA}, {CC, GFE},  
 {CC, CC}, {C/2C/2D/2D/2E/2E/2F/2F/2, CC}, {EDC, C/2C/2D/2D/2E/2E/2F/2F/2},  
 {CDE}  
 $d_2$ : {CCGG, CCEC}, {AAGG, FCEC}, {FFEE, DBCA}, {DD3/4E/4C2, FGC},  
 {CCGG, CCEC}, {AAGG, FCEC}, {FFEE, DBCA}, {DD3/4E/4C2, FGC}  
 $d_3$ : {FFGA}, {AGFE}, {DDEF}, {F3/2E/2E2}, {FFGA}, {AGFE}, {DDEF},  
 {E3/2D/2D2}

小節単位で出現音長をまとめ、その小節における小節構成音を決定する。この小節構成音が観測系列である(表4.2)。

小節	$d_1$	$d_2$	$d_3$
1	{C:1, D:1, E:1, F:1}	{C:5, E:1, G:2}	{F:2, G:1, A:1}
2	{C:2, D:2, E:2, F:1}	{C:2, E:1, F:1, G:2, A:2}	{E:1, F:1, G:1, A:1}
3	{C:1, D:1, E:2, F:1, G:1, A:1}	{C:1, D:1, E:2, F:2, A:1, B:1}	{D:2, E:1, F:1}
4	{E:2, F:2, G:2, A:1}	{C:13/4, D:7/4, E:1/4, F:1, G:1}	{E:5/2, F:3/2}
5	{C:2, E:1, F:1, G:1}	{C:5, E:1, G:2}	{F:2, G:1, A:1}
6	{C:4}	{C:2, E:1, F:1, G:2, A:2}	{E:1, F:1, G:1, A:1}
7	{C:3, D:1, E:1, F:1}	{C:1, D:1, E:2, F:2, A:1, B:1}	{D:2, E:1, F:1}
8	{C:2, D:2, E:2, F:1}	{C:13/4, D:7/4, E:1/4, F:1, G:1}	{D:5/2, E:3/2}
9	{C:1, D:1, E:1}		

表 4.1: 出現音

観測系列を用いて、ラベル付旋律について状態列となる平均律コード推定を行う。平均律コードの推定は、音楽理論に基づいて決定する(表4.3)。

状態列が決定することで、それぞれの旋律において状態遷移図が生成できる(図4.4, 図4.5)。

小節	1	2	3	4	5	6	7	8	9
$d_1$	DEF	CDE	EGA	EFG	CFG	C	CEF	CDE	CDE
$d_2$	C	CEG	CFA	CEG	DFB	CEA	DFG	C	
$d_3$	FGA	FGA	DEF	EF	FGA	FGA	DEF	EF	

表 4.2: 小節構成音：観測系列

小節	1	2	3	4	5	6	7	8	9
$d_1$	C	C	C	C	C	C	C	C	C
$d_2$	C	C	F	C	G	C	G	C	

表 4.3: 状態列：平均律コード

生成された状態遷移図に基づいて、 $d_3$  にクラス割り当てを行う。 $d_3$  のクラス所属確率は、状態遷移図の積計算で求めることが可能である。この時、各状態に所属する小節構成音も計算に用いられるが、状態の遷移が発生する際に小節構成音間の類似度も考慮することに注意を払う。以上の方法で  $d_3$  のクラス所属確率を求めると、それぞれ次の値となる。

前向きアルゴリズム (Forward Algorithm) :

$$P(A|d_3) = 4.006 \times 10^{-3}$$

$$P(B|d_3) = 2.216 \times 10^{-3}$$

ビタビ・アルゴリズム (Viterbi Algorithm) :

$$P(A|d_3) = 4.006 \times 10^{-3}$$

$$P(B|d_3) = 1.089 \times 10^{-3}$$

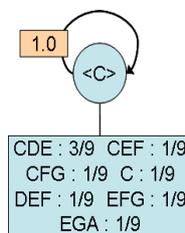


図 4.4: カエルの歌：状態遷移図

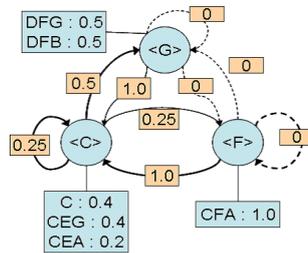


図 4.5: キラキラ星変奏曲：状態遷移図

この結果，いずれの場合も  $d_3$  が最も近いクラスは  $A$  であり，“歓喜の歌”は“カエルの歌”に割り当てられた．

## 4.5 実験

以下の実験では幾つかの変奏曲を扱う．変奏曲は1つのテーマと複数の変奏曲から構成されるため，ラベル付き訓練データはテーマの数だけあると考える．これ以外はラベルなしデータとみなし，旋律分類の対象とする．

### 4.5.1 準備

本稿では，複旋律で構成されたピアノ変奏曲を用いる．使用する楽曲は，Mozart キラキラ星変奏曲 K.265, Schubert 即興曲 Op.142-3, および Beethoven トルコ行進曲による6つの変奏曲 Op.76 の3つの変奏曲であり，それぞれ主題(テーマ)と12曲, 5曲, 6曲の変奏(バリエーション)からなる．これらは MIDI データとして表現されているものを ABC フォーマット [21] に変換して実験に使用する．

最初に，各楽曲をラベル付旋律集合とラベルなし旋律集合に分ける．ラベル付旋律集合は変奏曲テーマ(主旋律)3曲からなるとし，残り23の変奏曲はラベルなし旋律集合を構成する．各楽曲を表す ABC ファイルからラベル付旋律集合は全小節分の特徴量，ラベルなし旋律集合は先頭4小節分の特徴量，または先頭8小節分の特徴量を取り出す．

本実験で抽出する特徴量は，小節内の出現音の長さ上位3, 4, 5音を，その小節の小節構成音とする．

また，ラベル付旋律集合の特徴量に対して，それぞれの状態を示す平均律コードをあらかじめ設定する．ここで得られた状態を用いて，各ラベルなし旋律集合について隠れマルコフモデルを用いて分類，あるいはコード進行の付与をする．

### 4.5.2 評価方法

本実験では 3 つのクラス "Mozart", "Schubert", "Beethoven" を設定し, 隠れマルコフモデルを用いた分類器は全 23 変奏曲をいずれかのクラスに分類する. 変奏曲が "正しく分類" できたとはそのテーマがラベルの作曲者によって作曲されたときとする. 形式的に分類正解率を次のように定義する:

$$\frac{\text{正しく分類された変奏曲数}}{\text{総変奏曲数}}$$

本稿における総変奏曲数は 23 となる. また, 本研究では, HMM 状態を平均律コードと対応させるため, 変奏曲の各小節に対してコードが付与される. これは, 従来"コード進行"分析(アナリゼ)と呼ばれる問題に対する自動解の生成に相当する. 本研究で得られるコード進行と, 実際のコード進行との一致率を小節単位で評価する. 本実験では, 8 小節の特徴量に付与されたコードを用い, 分類結果で正解と判定された変奏曲を対象とする.

### 4.5.3 実験結果

状態列に平均律コードを用いた場合の先頭 4 小節を用いた際と先頭 8 小節を用いた際の生成確率を表 4.4 から表 4.7, 図 4.6 から図 4.13 に, そして, 各テーマに対する正解率を表 4.8 に, Viterbi アルゴリズムで付与されたコードの正解率を表 4.9 に示す.

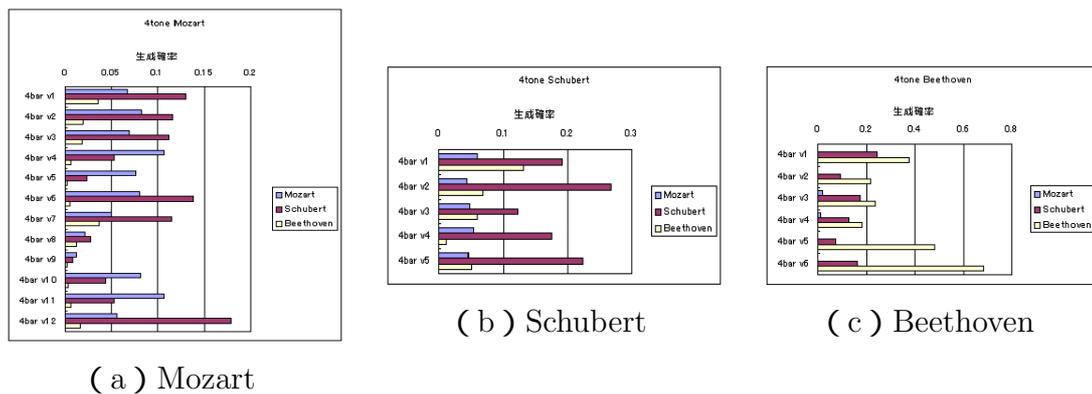


図 4.6: 生成確率: 4 音, 4 小節, Forward

テーマ判定に関しては, Viterbi アルゴリズムを用いた場合, 3 音の特徴量で小節数にかかわらず 100.0 %もの高い正解率を示した. コード判定に関しては, 3 音の特徴量で, 4 小節以上の一致が 69.6 %の正解率となった.

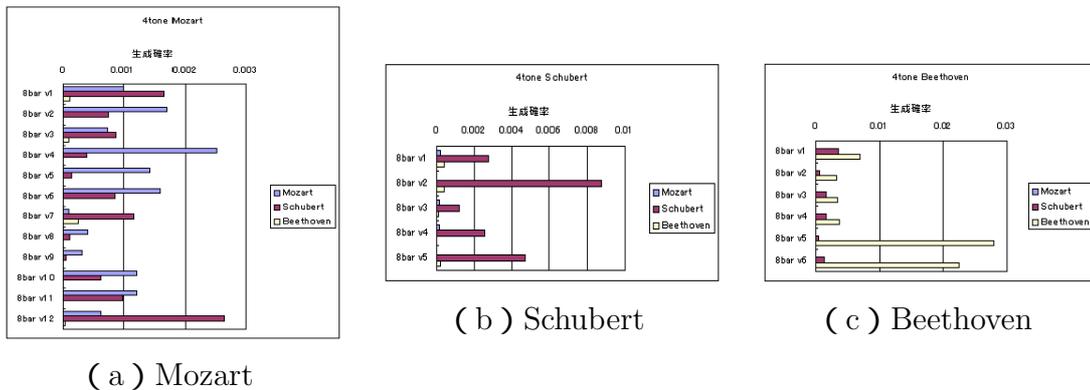


図 4.7: 生成確率: 4 音, 8 小節, Forward

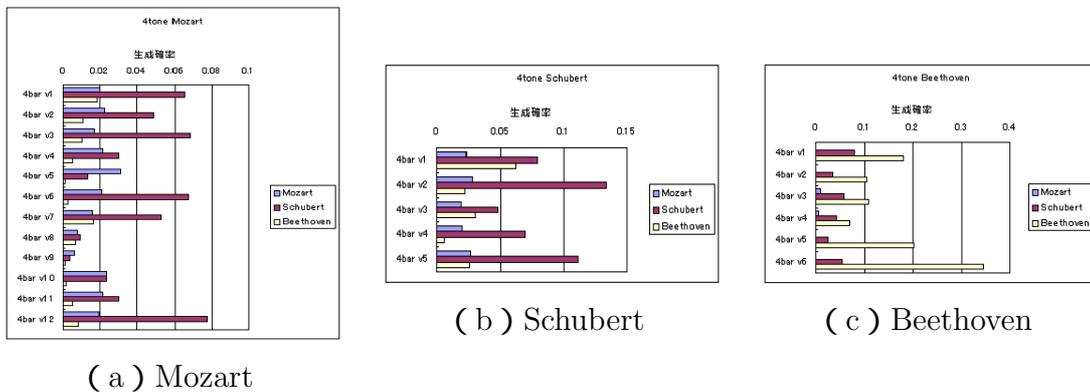


図 4.8: 生成確率: 4 音, 4 小節, Viterbi

#### 4.5.4 考察

分類正解率の結果では、特徴量の小節数を増やすことで分類正解率の向上が見られた。具体的には、3音の実験結果では、Viterbi アルゴリズムの場合で 95.7 % から 100.0 % に、4音の実験結果では、Forward アルゴリズムの場合で 69.6 % から 82.6 % に、Viterbi アルゴリズムの場合で 56.5 % から 78.3 % に正解率が向上した。しかし 5音の実験結果では、Forward アルゴリズムでは 39.1 % から 30.4 % と減少がみられた。この原因としていくつか考えられるが、小節構成音数が増加したことにより、他の変奏曲と大きく類似してしまう可能性がある。その結果として、小節数を増加するほど正解とは異なる変奏曲の生成確率が増大してしまい、誤判定をしたと考えられる。この問題は、以前の研究 [25] においても問題点としてあげている。

また、今回の実験では Forward アルゴリズムと Viterbi アルゴリズムの分類結果の

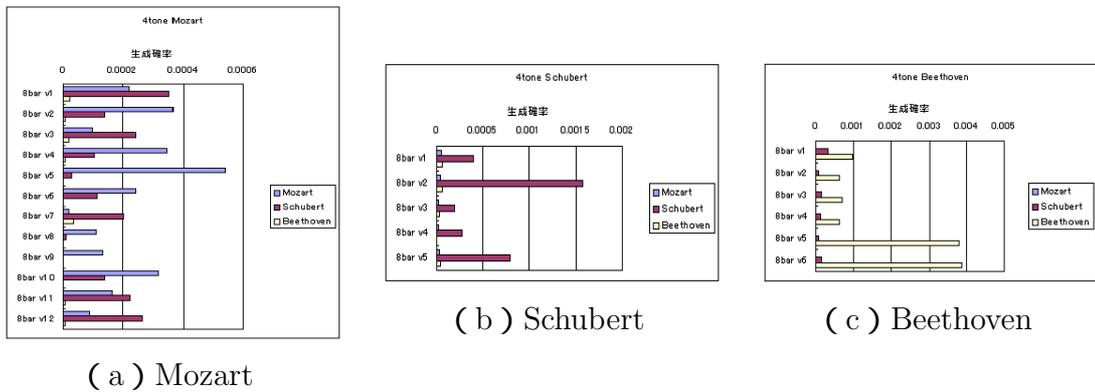


図 4.9: 生成確率: 4 音, 8 小節, Viterbi

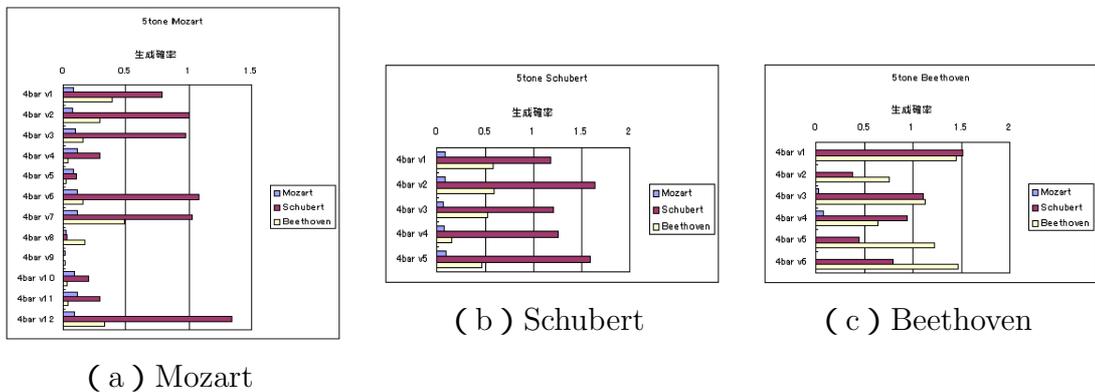


図 4.10: 生成確率: 5 音, 4 小節, Forward

比較も行った。特徴量が 3 音と 4 音に関しては、分類正解率の高さでは Forward アルゴリズムが高い値を示した。しかし、特徴量が 5 音に増加した場合には、Viterbi アルゴリズムが Forward アルゴリズムより高い分類率となった。また、Forward アルゴリズムでは小節数を増加させた結果、分類正解率の減少が起きたが、Viterbi アルゴリズムでは、向上まではいかないものの分類正解率を維持している。これは、アルゴリズムにおける計算手法の違いによるものと考えられる。Forward アルゴリズムで生成確率を計算する場合は、その値は全ての経路の総和になるのに対して、Viterbi アルゴリズムで生成確率を計算する場合は、最も確率の高い経路のみを選択することになる。今回扱った特徴量は出現音を組み合わせた形式であるが、その計算の中で互いの類似度を計算している。類似度計算で 0 確率を防ぐという利点がある一方、小節構成音数が増加していくにつれ互いに類似しあう音が増え、小節ごとにある特徴量が類似しやすく、あいまいさが増してしまう。さ

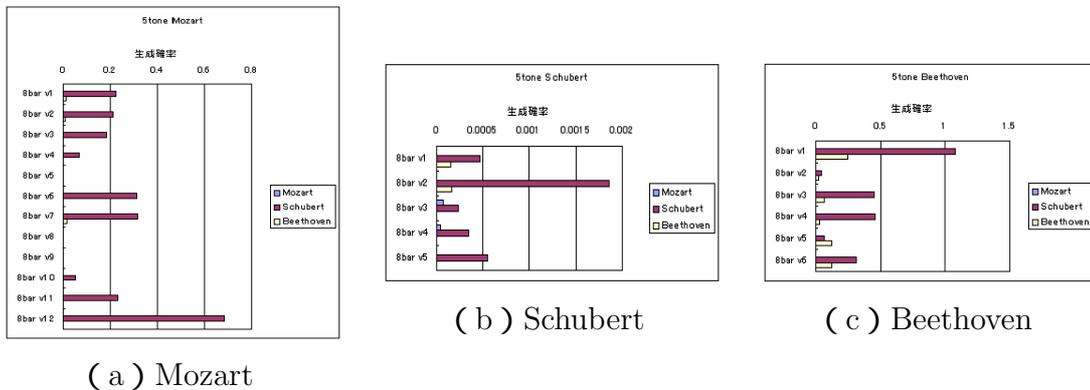


図 4.11: 生成確率: 5 音, 8 小節, Forward

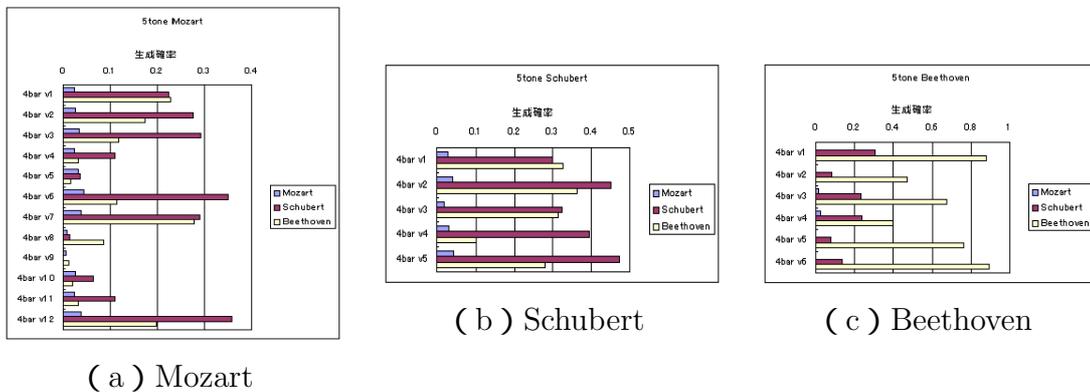


図 4.12: 生成確率: 5 音, 4 小節, Viterbi

らに, Forward アルゴリズムのように総経路をたどった場合には, 本来とは異なる小節構成音の確率が大幅に増大する可能性が考えられる. このことから, 本研究で用いたような音を組み合わせた形式による特徴量を扱う上では, 音数が少量で構成される場合は Forward アルゴリズム, 多量の音数で構成される場合は Viterbi アルゴリズムが望ましいと考えられる.

正解率の比較として, 単旋律曲の分類において最も正解率の高かった特徴量を先頭 4 小節でピッチスペクトラムを用い, 分類手法に単純ベイズ, および EM アルゴリズムを用いた場合の正解率 [23] を表 4.10 に示す.

単純ベイズ法 (EM loop 0 回) で最高 87 %, EM アルゴリズムで最高 91.3 %を示したが, 本研究の手法においても, 単旋律から複旋律へと複雑化しているにもかかわらず特徴量抽出を 8 小節用いた場合の正解率が 100.0 %と, 単旋律での分類と比較しても互角以上の正解率となった. 本研究の手法は, 複旋律分類におい

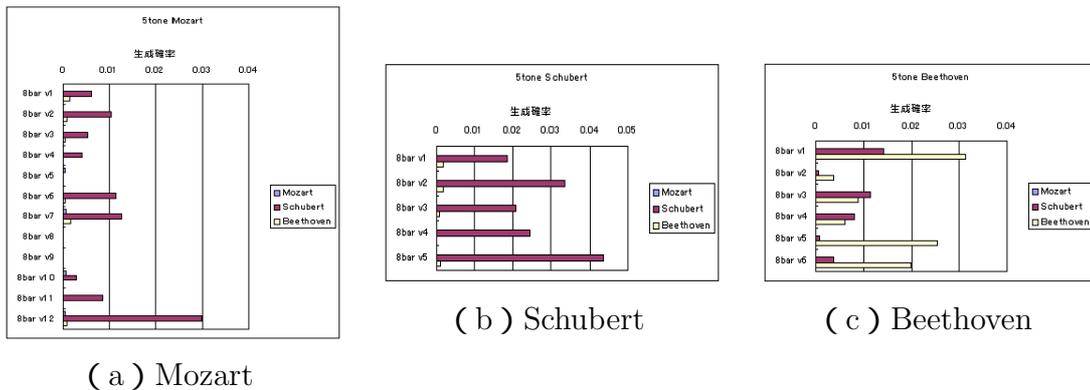


図 4.13: 生成確率: 5 音, 8 小節, Viterbi

て有効な抽出方法といえよう。

コード判定におけるコード一致率については, 多くの変奏曲において完全一致がないものの, 特徴量の半数にあたる 4 小節以上の一致率は 55.6 % から 69.6 % という 7 割近い変奏曲において一致が見られた. 不一致を起こした原因として, 変奏曲ごとに一定の箇所における不一致が多く見られた. 具体的には Mozart 変奏曲では, 多くの場合先頭小節のコードに不一致が見られた. これにより, 完全一致の割合が大きく減少しているものと考えられる. また Schubert 変奏曲では, 5 小節から 8 小節間において正解となるコード進行の遷移がテーマになく消滅していることで, 異なるコード進行を選択した結果, 最終的なコードの不一致が見られた. これにより, 一致率の低下に繋がったと考えられる. しかし, 多少の誤差を考慮したうえでのコード割当は, 計算により求められることが可能であり, 有効な手段であるといえよう.

## 4.6 結び

本研究では, あらかじめクラスが与えられた少量の複旋律楽曲から多く存在する未知旋律を分類する方法を提案した. 具体的には, 複旋律情報の楽曲を小節単位の音情報から構成した特徴量も用いることと, 隠れマルコフモデルを用いてモデル推定を行い, 生成されたモデルで計算することにより, 分類結果で高い正解率が得られた. この結果, 複旋律情報においても高い分類精度を得ることを確認した. また, 小節構成音と少量の複旋律楽曲にあらかじめコードを付与することで, 多くの未知旋律に新たにコードを付与する方法も提案した. この結果, 半数以上の小節において高い一致率を得ることを確認した.

	Mozart	Schubert	Beethoven
ozart v1	$6.627 \times 10^{-2}$	$2.882 \times 10^{-2}$	$3.901 \times 10^{-3}$
v2	$3.389 \times 10^{-2}$	$3.316 \times 10^{-2}$	$2.231 \times 10^{-3}$
v3	$9.266 \times 10^{-2}$	$6.089 \times 10^{-3}$	$2.797 \times 10^{-4}$
v4	$9.266 \times 10^{-2}$	$6.089 \times 10^{-3}$	$2.797 \times 10^{-4}$
v5	$7.355 \times 10^{-2}$	$2.799 \times 10^{-3}$	$2.474 \times 10^{-4}$
v6	$6.311 \times 10^{-2}$	$9.422 \times 10^{-3}$	$2.253 \times 10^{-4}$
v7	$6.133 \times 10^{-2}$	$1.950 \times 10^{-2}$	$1.312 \times 10^{-3}$
v8	$1.292 \times 10^{-2}$	$4.068 \times 10^{-3}$	$2.559 \times 10^{-3}$
v9	$1.117 \times 10^{-2}$	$5.595 \times 10^{-3}$	$7.388 \times 10^{-4}$
v10	$9.266 \times 10^{-2}$	$6.089 \times 10^{-3}$	$2.797 \times 10^{-4}$
v11	$9.266 \times 10^{-2}$	$6.089 \times 10^{-3}$	$2.797 \times 10^{-4}$
v12	$4.840 \times 10^{-2}$	$2.222 \times 10^{-2}$	$7.638 \times 10^{-4}$
Schubert v1	$1.769 \times 10^{-2}$	$5.584 \times 10^{-2}$	$4.634 \times 10^{-2}$
v2	$1.199 \times 10^{-2}$	$1.403 \times 10^{-1}$	$4.859 \times 10^{-2}$
v3	$4.036 \times 10^{-2}$	$8.897 \times 10^{-2}$	$8.396 \times 10^{-3}$
v4	$3.391 \times 10^{-2}$	$5.263 \times 10^{-2}$	$6.980 \times 10^{-4}$
v5	$1.549 \times 10^{-2}$	$7.399 \times 10^{-2}$	$2.529 \times 10^{-3}$
Beethoven v1	$8.919 \times 10^{-4}$	$1.773 \times 10^{-2}$	$3.1070 \times 10^{-1}$
v2	$3.211 \times 10^{-3}$	$2.902 \times 10^{-2}$	$4.774 \times 10^{-2}$
v3	$5.077 \times 10^{-3}$	$3.037 \times 10^{-2}$	$1.872 \times 10^{-1}$
v4	$1.508 \times 10^{-3}$	$7.753 \times 10^{-3}$	$6.684 \times 10^{-2}$
v5	$1.522 \times 10^{-3}$	$1.473 \times 10^{-2}$	$2.592 \times 10^{-1}$
v6	$3.686 \times 10^{-4}$	$4.752 \times 10^{-2}$	$5.285 \times 10^{-1}$

表 4.4: 生成確率: 3 音, 4 小節, Forward

	Mozart	Schubert	Beethoven
Mozart v1	$5.048 \times 10^{-4}$	$2.766 \times 10^{-4}$	$6.936 \times 10^{-7}$
v2	$2.655 \times 10^{-4}$	$4.119 \times 10^{-5}$	$1.325 \times 10^{-7}$
v3	$8.930 \times 10^{-4}$	$7.280 \times 10^{-6}$	$7.763 \times 10^{-8}$
v4	$4.683 \times 10^{-4}$	$7.563 \times 10^{-7}$	$2.109 \times 10^{-7}$
v5	$1.299 \times 10^{-3}$	$5.055 \times 10^{-6}$	$7.259 \times 10^{-8}$
v6	$5.186 \times 10^{-4}$	$1.010 \times 10^{-5}$	$2.453 \times 10^{-7}$
v7	$1.168 \times 10^{-4}$	$2.825 \times 10^{-5}$	$2.613 \times 10^{-6}$
v8	$7.936 \times 10^{-5}$	$2.646 \times 10^{-6}$	$3.523 \times 10^{-8}$
v9	$9.991 \times 10^{-5}$	$3.015 \times 10^{-6}$	$1.412 \times 10^{-7}$
v10	$9.513 \times 10^{-4}$	$2.377 \times 10^{-5}$	$9.096 \times 10^{-7}$
v11	$1.978 \times 10^{-3}$	$1.084 \times 10^{-5}$	$2.485 \times 10^{-9}$
v12	$1.078 \times 10^{-3}$	$4.747 \times 10^{-5}$	$8.706 \times 10^{-7}$
Schubert v1	$5.378 \times 10^{-6}$	$4.738 \times 10^{-4}$	$1.519 \times 10^{-4}$
v2	$2.884 \times 10^{-6}$	$1.854 \times 10^{-3}$	$1.643 \times 10^{-4}$
v3	$7.262 \times 10^{-5}$	$2.381 \times 10^{-4}$	$5.609 \times 10^{-7}$
v4	$3.996 \times 10^{-5}$	$3.424 \times 10^{-4}$	$1.892 \times 10^{-7}$
v5	$4.087 \times 10^{-6}$	$5.485 \times 10^{-4}$	$8.445 \times 10^{-6}$
Beethoven v1	$6.442 \times 10^{-8}$	$1.453 \times 10^{-5}$	$3.165 \times 10^{-3}$
v2	$2.094 \times 10^{-7}$	$3.980 \times 10^{-5}$	$1.796 \times 10^{-4}$
v3	$4.006 \times 10^{-6}$	$3.261 \times 10^{-5}$	$7.940 \times 10^{-5}$
v4	$4.548 \times 10^{-8}$	$6.053 \times 10^{-6}$	$8.211 \times 10^{-4}$
v5	$1.042 \times 10^{-7}$	$1.320 \times 10^{-5}$	$8.217 \times 10^{-3}$
v6	$8.681 \times 10^{-9}$	$1.434 \times 10^{-4}$	$1.894 \times 10^{-2}$

表 4.5: 生成確率: 3音, 8小節, Forward

	Mozart	Schubert	Beethoven
Mozart v1	$2.007 \times 10^{-2}$	$1.791 \times 10^{-2}$	$2.435 \times 10^{-3}$
v2	$9.635 \times 10^{-3}$	$1.257 \times 10^{-2}$	$1.124 \times 10^{-3}$
v3	$2.572 \times 10^{-2}$	$3.243 \times 10^{-3}$	$2.341 \times 10^{-4}$
v4	$2.572 \times 10^{-2}$	$3.243 \times 10^{-3}$	$2.341 \times 10^{-4}$
v5	$3.150 \times 10^{-2}$	$1.390 \times 10^{-3}$	$2.150 \times 10^{-4}$
v6	$3.536 \times 10^{-2}$	$4.918 \times 10^{-3}$	$1.714 \times 10^{-4}$
v7	$3.074 \times 10^{-2}$	$9.694 \times 10^{-3}$	$4.724 \times 10^{-4}$
v8	$5.281 \times 10^{-3}$	$1.656 \times 10^{-3}$	$1.623 \times 10^{-3}$
v9	$6.166 \times 10^{-3}$	$2.895 \times 10^{-3}$	$7.388 \times 10^{-4}$
v10	$2.572 \times 10^{-2}$	$3.243 \times 10^{-3}$	$2.341 \times 10^{-4}$
v11	$2.572 \times 10^{-2}$	$3.243 \times 10^{-3}$	$2.341 \times 10^{-4}$
v12	$1.421 \times 10^{-2}$	$1.129 \times 10^{-2}$	$3.142 \times 10^{-4}$
Schubert v1	$1.073 \times 10^{-2}$	$2.715 \times 10^{-2}$	$2.035 \times 10^{-2}$
v2	$6.953 \times 10^{-3}$	$7.175 \times 10^{-2}$	$2.815 \times 10^{-2}$
v3	$1.365 \times 10^{-2}$	$4.380 \times 10^{-2}$	$4.257 \times 10^{-3}$
v4	$9.786 \times 10^{-3}$	$2.132 \times 10^{-2}$	$4.296 \times 10^{-4}$
v5	$9.481 \times 10^{-3}$	$3.753 \times 10^{-2}$	$1.272 \times 10^{-3}$
Beethoven v1	$4.199 \times 10^{-4}$	$5.182 \times 10^{-3}$	$1.329 \times 10^{-1}$
v2	$2.084 \times 10^{-3}$	$9.048 \times 10^{-3}$	$2.086 \times 10^{-2}$
v3	$2.959 \times 10^{-3}$	$1.011 \times 10^{-2}$	$8.609 \times 10^{-2}$
v4	$1.214 \times 10^{-3}$	$4.548 \times 10^{-3}$	$2.834 \times 10^{-2}$
v5	$9.885 \times 10^{-4}$	$5.067 \times 10^{-3}$	$9.776 \times 10^{-2}$
v6	$1.839 \times 10^{-4}$	$1.436 \times 10^{-2}$	$2.693 \times 10^{-1}$

表 4.6: 生成確率: 3 音, 4 小節, Viterbi

	Mozart	Schubert	Beethoven
Mozart v1	$1.381 \times 10^{-4}$	$9.486 \times 10^{-5}$	$1.883 \times 10^{-7}$
v2	$4.592 \times 10^{-5}$	$5.253 \times 10^{-6}$	$3.053 \times 10^{-8}$
v3	$2.029 \times 10^{-4}$	$1.739 \times 10^{-6}$	$3.411 \times 10^{-8}$
v4	$7.926 \times 10^{-5}$	$1.790 \times 10^{-7}$	$1.285 \times 10^{-7}$
v5	$5.239 \times 10^{-4}$	$1.205 \times 10^{-6}$	$2.489 \times 10^{-8}$
v6	$1.979 \times 10^{-4}$	$3.514 \times 10^{-6}$	$7.536 \times 10^{-8}$
v7	$3.293 \times 10^{-5}$	$3.905 \times 10^{-6}$	$3.714 \times 10^{-7}$
v8	$2.132 \times 10^{-5}$	$3.558 \times 10^{-7}$	$1.301 \times 10^{-8}$
v9	$3.726 \times 10^{-5}$	$4.417 \times 10^{-7}$	$8.013 \times 10^{-8}$
v10	$2.344 \times 10^{-4}$	$6.028 \times 10^{-6}$	$3.861 \times 10^{-7}$
v11	$4.790 \times 10^{-4}$	$2.549 \times 10^{-6}$	$1.708 \times 10^{-9}$
v12	$2.515 \times 10^{-4}$	$9.577 \times 10^{-6}$	$1.490 \times 10^{-7}$
Schubert v1	$2.955 \times 10^{-6}$	$7.940 \times 10^{-5}$	$5.691 \times 10^{-5}$
v2	$1.503 \times 10^{-6}$	$4.015 \times 10^{-4}$	$5.194 \times 10^{-5}$
v3	$2.746 \times 10^{-6}$	$4.152 \times 10^{-5}$	$1.165 \times 10^{-7}$
v4	$7.251 \times 10^{-6}$	$2.975 \times 10^{-5}$	$8.344 \times 10^{-8}$
v5	$2.226 \times 10^{-6}$	$9.481 \times 10^{-5}$	$3.326 \times 10^{-6}$
Beethoven v1	$1.150 \times 10^{-8}$	$9.574 \times 10^{-7}$	$5.460 \times 10^{-4}$
v2	$6.571 \times 10^{-8}$	$3.871 \times 10^{-6}$	$2.869 \times 10^{-5}$
v3	$1.863 \times 10^{-6}$	$2.741 \times 10^{-6}$	$1.424 \times 10^{-5}$
v4	$2.747 \times 10^{-8}$	$6.183 \times 10^{-7}$	$2.191 \times 10^{-4}$
v5	$3.606 \times 10^{-8}$	$1.869 \times 10^{-6}$	$1.064 \times 10^{-3}$
v6	$1.730 \times 10^{-9}$	$1.044 \times 10^{-5}$	$3.529 \times 10^{-3}$

表 4.7: 生成確率: 3音, 8小節, Viterbi

( ) 内は小節構成音数	使用モデル	
	Chord(Forward)	(Viterbi)
(3) 4小節	100.0 %	95.7 %
(3) 8小節	100.0 %	100.0 %
(4) 4小節	69.6 %	56.5 %
(4) 8小節	82.6 %	78.3 %
(5) 4小節	39.1 %	47.8 %
(5) 8小節	30.4 %	47.8 %

表 4.8: 正解率 / テーマ判定

	正解小節数		
	8小節	6小節以上	4小節以上
3音	4.35 %	39.1 %	69.6 %
4音	0.0 %	38.9 %	55.6 %
5音	0.0 %	45.5 %	63.6 %

表 4.9: 正解率 / コード判定

		EM loop					
		0	5	10	15	20	25
類似度 (%)	30	87	91.3	91.3	91.3	91.3	91.3
	50	78.3	91.3	91.3	87.0	87.0	87.0

表 4.10: 正解率 : 単旋律 , ピッチスペクトル

## 第5章 結論

本研究では、旋律情報から特徴量を抽出し、確率的手法を用いて分類する方法を提案し、実験によりその有効性を証明した。

単旋律情報を分類において、いくつかの特徴量を比較し、EM アルゴリズムにより分類する方法を提案し、高い分類正解率を得ることができた。

次に、複旋律情報の分類において、複旋律情報を扱える特徴量を提案し、マルコフモデルを用いて高い分類正解率を得ることができた。

コード進行を用いた分類において、複旋律情報での分類正解率がより高いものを得ることができ、さらに、未知旋律集合に正しいコード進行を付与することができる可能性を示した。

また、全体を通じて有効な特徴量の一定の傾向を見ることが出来た。具体的には、楽譜形式の旋律を用いた場合では、3 音程度の和音形式が一番高い分類正解率を得ることができ、計算に用いる旋律特徴量を 4 小節から 8 小節と増加させることで、正解率が改善されるケースが多くの場合で見られた。

今後の課題として、特徴量における拍子等時間軸の伸張性を持たせることが必要である。小節単位で構成する特徴量が多く、拍子が変わった場合にうまく対応できないケースがよく見られた。特徴量が少ないケースでは、改善のための有効な手段になると考えられる。関連して、誤差を含む類似性を考慮することも考えられる。具体的には、0 確率を避けるために用いている類似度があるが、ラベルに存在する特徴量の種類や確率が平均的に現れてしまうと、他のラベルの正解に変化してしまう可能性がみられた。0 確率を類似度に置き換えずに無視することで、同様に和音進行においても、一定の間違いを生む区間の特徴量を無視することで、より多くの正解が得られる可能性がある。しかし、無視された特徴量が多くなると、正解を得るために必要な特徴量が選ばれなくなる可能性も出てくるため、手法も含めて検討することが必要と思われる。

また、本研究では触れていないが、音楽理論における例外処理などの条件を加えることも有効である可能性がある。しかし、条件設定の有効性についても、特徴量を抽出した段階で元の音楽理論とは異なる形式となるため、有効性についての検証・議論が必要となるとと思われる。

## 謝辞

本研究に遂行するにあたり，多くの方々にお世話になりました．この場を借りて，感謝の意を述べさせていただきたいと思います．特に，日頃より数々のご指導をいただいた指導教官である，法政大学工学部情報電気電子工学科 三浦孝夫 教授に心から深く御礼を申し上げます．

また，産業能率大学経営情報学部 塩谷勇 教授にも多くのご指導をいただきました．心から感謝申し上げます．

データ工学研究室の先輩方，同輩，後輩たちにも，研究活動から学生生活まで幅広いお付き合いで大変お世話になりました．

最後になりましたが，このような形で研究をまとめることができたのは，両親をはじめ，多くの皆様方の御支援，御協力の賜物であります．この場をお借りして，深く御礼を申し上げます．

## 関連図書

- [1] Dowling W.J.: Scale and Contour – two components of a theory of memory for melodies, *Psychological Reviews* 85-4, 1978, pp.341-354
- [2] Droettboom, M. et al.: An Approach Towards A Polyphonic Music Retrieval System, *Intn'l Symp. on Music Information Retrieval (ISMIR)*, 2001
- [3] Droettboom, M. et al.: Expressive and Efficient Retrieval of Symbolic Musical Data, *Intn'l Symp. on Music Information Retrieval (ISMIR)*, 2002
- [4] Ghais, A. et al.: Query By Humming, *ACM Multimedia Conf.*, 1995
- [5] Grossman, D., O'.Frieder: Information Retrieval – Algorithms and Heuristics, Kluwer Academic Press, 1998
- [6] Kageyama, T. et al.: Melody Retrieval With Humming, *ACM Intn'l Computer Music Conf. (ICMC)*, 1993
- [7] Kim, Y. et al.: Analysis of A Contour-based Representation for Melody, *Intn'l Symp. on Music Information Retrieval (ISMIR)*, 2000
- [8] Lie, L. et al.: A New Approach to Query By Humming in Music Retrieval, *Intn'l Conf on Multimedia and Expo (ICME)*, 2001
- [9] Meek, C. et al.: Thematic Extractor, *Intn'l Symp. on Music Information Retrieval (ISMIR)*, 2001
- [10] Miura, T. and Shioya, I.: Similarities among Melodies for Music Information Retrieval, *ACM Conf. on Information and Knowledge Management(CIKM)*, 2003
- [11] Mitchell, T.: Machine Learning, McGrawHill Companies, 1997

- [12] Nigam, K. McCallum, A. Thrun, I. Mitchell, T.: Text Classification from Labeled and Unlabeled Documents using EM, Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.
- [13] Pickens, J.: A Comparison of Language Modelling and Probabilistic Text Information Retrieval Approaches to Monophonic Music Retrieval, *Intn'l Symp. on Music Information Retrieval (ISMIR)*, 2000
- [14] Pickens, J. and Crawford, T.: Harmonic Models for Polyphonic Music Retrieval, *ACM Conf. on Information and Knowledge Management (CIKM)*, 2002
- [15] Ramakrishnan, R. and Gehrke, J.: Database Management Systems (2nd ed) 2000, McGrawHill
- [16] Sonoda, T. et al.: A WWW-based Melody Retrieval System, *Intn'l Computer Music Conf. (ICMC)*, 1998
- [17] Uitdenbogerd, A.L. et al.: Manipulation of Music For Melody Matching, *ACM MultiMedia Conf.*, 1998
- [18] Uitdenbogerd, A.L. et al.: Matching Techniques for Large Music Databases, *ACM Multimedia Conf.*, 1999
- [19] Uitdenbogerd, A.L. et al.: MUSIC IR: Past, Present and Future, *Intn'l Symp. on Music Information Retrieval (ISMIR)*, 2000
- [20] Uitdenbogerd, A.L. et al.: Music Ranking Techniques Evaluated, *Intn'l Symp. on Music Information Retrieval (ISMIR)*, 2000
- [21] Walshaw, C.: abc Version 1.6,  
<http://www.gre.ac.uk/~c.walshaw/abc2mtex/abc.txt>
- [22] Yang, C.: Music Database Retrieval Based on Spectral Similarity, *Intn'l Symp. on Music Information Retrieval (ISMIR)*, 2001
- [23] Yoshihara, Y. and Miura, T: Melody Classification Using EM Algorithm. *Computer Software and Applications Conference (COMPSAC)*, pp. 204-210, 2005
- [24] Yoshihara, Y. and Miura, T: Classifying Polyphonic Music Based on Markov Model. *Intelligent Data Engineering and Automated Learning (IDEAL)*, pp. 697-706, 2006

- [25] 吉原 幸輝, 三浦 孝夫.: 確率過程に基づく複旋律分類, 日本データベース学会 Letters Vol.5, No.1, pp. 137-140, 2006
- [26] 石桁他: 新装楽典 - 理論と実習, 音楽之友社, 2001
- [27] 岩崎学: 不完全データの統計解析, エコノミクス社, 2002
- [28] 上嶋 宏, 三浦 孝夫, 塩谷 勇.: 同義語, 多義語の考慮による文書分類の精度向上, 電子情報通信学会誌 Vol.J87-D-I No.2, 2004
- [29] 北研二他: 情報検索のアルゴリズム, 共立出版, 2002
- [30] 新納 浩幸, 佐々木 捻.: EM アルゴリズムの最適ループ回数の予測を用いた語義判別規則の教師なし学習, 情報処理学会論文誌 Vol.44, No.12, 2003