Vol.2017-MUS-116 No.19
2017/8/26

作曲者判定タスクのために分析すべき楽曲の長さ

高本 綺架[1,a)]　吉田 光男[1,b)]　梅村 恭司[1,c)]　市川 裕子[2,d)]

概要：楽曲の楽譜情報を利用し，作曲者を判定する取り組みが行われている．このような取り組みでは，楽曲を文字列に変換し，その情報量を計算して作曲者を判定している．本研究では，作曲者判定において作曲者の特徴として捉えられているパターンに着目し，その特徴がどの範囲で現れるのかを分析した．従来はある楽曲に含まれる情報量の計算に BZIP2 などの圧縮プログラムを用いていたが，圧縮プログラムでは楽曲において特徴として捉えられている遷移の長さの特定は難しい．そこで我々は，文字列に含まれる情報量を計算するプログラムを構築し，情報量を計算する際の文字列の長さに対し制限をかけることを実現した．このプログラムを用いて，情報量を計算する際に用いる音の長さを特定の長さに制限し，作曲者判定を行なった．判定結果の正解数から，作曲者を判定するために必要な音の遷移の長さを分析した結果，実際にはより長いパターンを特定できればより有用であると考えていたが，隣り合った 2 つの音の遷移に相当する区間の分析で作曲者を判定できることがわかった．

キーワード：音楽情報処理，作曲者判定，情報量

A Study for the Required Length of Music Pieces to Analyze for Composer Estimation Task

Ayaka Takamoto[1,a)]　Mitsuo Yoshida[1,b)]　Kyoji Umemura[1,c)]　Yuko Ichikawa[2,d)]

1. はじめに

人は音楽を聴いたとき，様々な情報を認識できる．例えば，ジャンルや作曲者が挙げられる．ジャンルの判定は和音の進行などによって判定できると思われるが，作曲者の判定はどこに作曲者の情報が現れるかの認識が難しく，またその情報がどこにあるのかは興味深い問題である [1], [2], [3]．この作曲者を判定するというタスクについて，楽曲の楽譜情報を音の有無を表現する形式に変更し判定する手法がある [4]．本研究では，この作曲者判定を用いて，楽曲における遷移の長さについて作曲者の特徴がどの程度の長さに出現するのかを分析する．

先行研究 [5] における作曲者判定では，作曲者の特徴を捉えるために圧縮プログラムを用いている．楽曲を文字列化し，その文字列を圧縮して楽曲に含まれる情報量を計算する．これは圧縮プログラムに存在する繰り返しなどによる共通のパターンが多いほど圧縮率が高くなるという性質を利用している．しかし，既存の圧縮プログラムを利用した場合では，作曲者の特徴として捉えられているパターンの長さを調整しての分析は困難である．そこで我々は，圧縮した際のファイルサイズは文字列の情報量の近似と考え，文字列の持つ情報量を情報理論に基づき計算するプログラムを作成し，この情報量で分析した．これを用いて楽曲の持つ情報量を計算する際，遷移の長さに制限をかけて作曲者を判定し，判定の正解数から作曲者の持つ特徴を分析する．

[1] 豊橋技術科学大学
[2] 東京工業高等専門学校
[a)] a153350@edu.tut.ac.jp
[b)] yoshida@cs.tut.ac.jp
[c)] umemura@tut.jp
[d)] yuko@tokyo-ct.ac.jp

2. 作曲者判定

2.1 楽曲の文字列化

作曲者を判定するためには，対象となる楽曲データの文

字列化が必要になる．本節では，楽曲の文字列化手法について述べる．これは先行研究 [5] と同様である．

本研究で対象とする楽曲はピアノ曲であるため，各鍵盤をセンサと考える．楽曲データである MIDI ファイルには，音が鳴り始めたタイミングと鳴り終わるタイミングが記録されている．この情報をノート情報と呼び，ノート情報を用いてある時間における音のオン/オフ情報を取得する．ある楽曲データにおいて音が鳴っている時は 1，鳴っていない時は 0 という文字を全ての鍵盤に当てはめることで要素数 88 のベクトルを生成する．このベクトルをピッチベクトルとする．ある時刻 T における鍵盤の状態を図 1 に示す．なお，図 1 において編みかけの部分は鳴っている鍵盤を表しており，1 として文字列化され，編みかけが無い箇所は鳴っていない鍵盤を表しているため 0 として文字列化される．ここから生成されるピッチベクトル p は図 2 のようになる．図 1 において p[38] などは，最も低い鍵盤を 0 番目とした時，何番目の鍵盤であるかを示している．

楽曲を文字列化する際，音符が変化した際にノート情報を抽出する方法と，一定の時間間隔で抽出する方法が考えられる．音符が変化した点で抽出すると，楽曲が持つリズム情報が失われてしまうという問題が生じる．従って，本研究では一定の時間間隔で抽出し文字列化する．この時間間隔を実時間にした場合，楽曲のテンポによって抽出できる情報に差が出てしまう．そのため，抽出間隔を 4 文音符などの相対時間として設定する．相対時間を用いて情報を抽出すると，テンポ情報が失われてしまうが，テンポの異なる楽曲から一定の情報を取得できる．本研究では，この相対時間感覚を先行研究において最も高精度であった 16 分音符単位と設定する [4].

上記の手法を用いて実際の楽曲から文字列を取得する．図 3 のような楽譜情報を持つ MIDI ファイルからノート情報を抽出し，時間順にソートする．ソートされたデータから 16 分音符間隔で音のオンオフ情報を取得しピッチベクトルを生成する．このように生成したピッチベクトルを直前のデータの終端に連結すると図 4 のような 0 と 1 のみから構成される長い文字列を得ることができる．

この連結において，ピッチベクトル同士の間には区切りとなる文字は挿入しない．全てのピッチベクトルをそのまま連結することで，楽曲内で転調が発生した時にも旋律の同一性が判定できる．図 5 において，左はハ長調，右はト長調である．この 2 つの楽曲は一見違うものに見えるが，ある旋律がシフトしたものと考えることもできる．各調の楽曲を区切り文字ありで文字列化したものを図 6 に，区切り文字でなしで文字列化したものを図 7 に示す．図 6 のように区切り文字を挿入せず文字列化した場合，下線部の文字はハ長調，ト長調で同一の文字列となる．しかし，区切り文字を使用して文字列化した場合，図 7 に示すように区切り文字のために同一文字列と判定されない．以上の理由

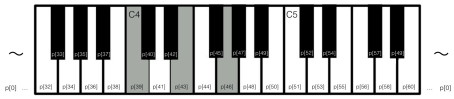

図 1 ある時刻のオン/オフ情報

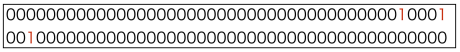

図 2 ある時刻のピッチベクトル

図 3 ある楽曲とそのピッチベクトルの一部

図 4 楽曲を文字列化した結果

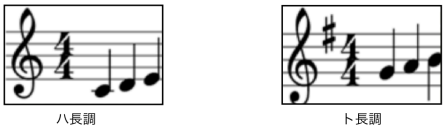

図 5 ハ長調とト長調

から，各ピッチベクトルの間には区切り文字を挿入せず文字列化する．

2.2 作曲者の判定方法

作曲者判定は，文字列化した未知楽曲の情報量を既知の楽曲群の情報を用いて計算することで行う．判定の様子を図 8 に示す．これは先行研究 [4], [5] とは異なる．図におい

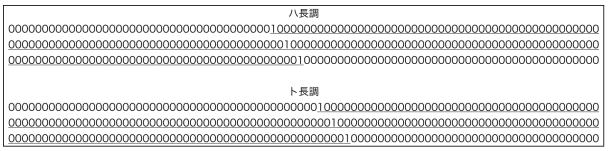

図 6 区切り文字なしでの文字列化

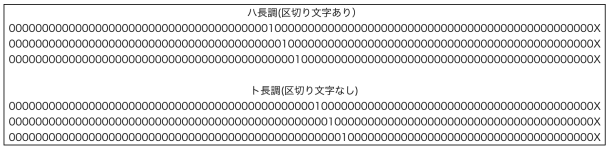

図 7 区切り文字ありでの文字列化

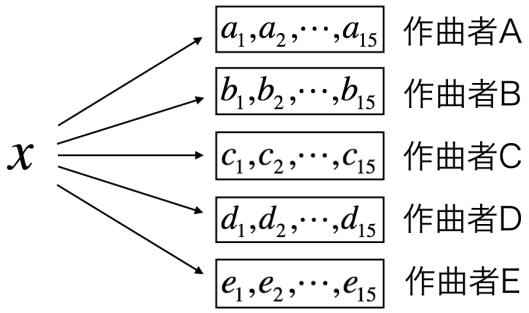

図 8 グループ化を用いた判定方法

て x は未知の楽曲を表し，a_1,a_2,...,a_{15} などの集合は作曲者 A の楽曲全てを連結したものである．未知曲 x に対して判定を行う場合，各作曲者 A,B,C,D,E の情報を用いて x の情報量を計算する．算出された 5 つの情報量のうち，最も情報量が小さい楽曲群の作曲者を，未知曲の作曲者と判定する．

3. 情報量の計算方法

この節では，情報理論に基づいて文字列に含まれる情報量を計算する手法について検討する．一般にある事象の情報量は，それが起きる確率 P に依存して決まる [6]．従って文字列の中で，ある文字 c が出現する確率が $P(c)$ であったとき，その情報量 $I_c(c)$ は式 (1) のように表せる．

$$I_c(c) = -\log_2 P(c) \tag{1}$$

文字列が一文字単位で構成される系列と考えると，その情報量は一文字あたりの情報量 $I_c(c)$ の総和と等しい．長さ N の文字列 S に含まれる互いに独立な i 番目の文字を c_i，その生起確率を $P(c_i)$ とした場合，その情報量 $I_c(S)$ は式 (2) のようになる．

$$\begin{aligned} I_c(S) &= -\log_2(\prod_{i=1}^{N} P(c_i)) \\ &= -\sum_{i=1}^{N} \log_2 P(c_i) \end{aligned} \tag{2}$$

文字列 “abc”であれば，この文字列は “a”,“b”,“c”から構成される系列と捉えることができる．この時，各文字の生起確率が 1/2，1/4，1/2 とすると，各文字の情報量は $-\log_2 1/2$，$-\log_2 1/4$，$-\log_2 1/2$ である．各文字が互いに独立ならば，文字列 “abc”の情報量は各文字の情報量の総和と等しいため，$I(\text{“}abc\text{”}) = I(\text{“}a\text{”}) + I(\text{“}b\text{”}) + I(\text{“}c\text{”}) = -\log_2 1/2 - \log_2 1/4 - \log_2 1/2 = 4$ となる．

しかし，実際の文字列では単語のような特定のパターンが繰り返し出現する場合が多い．従って，文字列を特定のパターンからなる系列であると考えると，その情報量 $I_s(S)$ は式 (3) のように表すことができる．

$$I_s(S) = \min_{\pi_s \in \pi(S)} \Big(-\sum_{t \in \pi_s} \log_2 P(t) \Big) \tag{3}$$

$\pi(S)$ は S の 2^{N-1} 通りの分割方法の集合であり，π_s は分割された文字列の集合である．また，π_s の要素を t とする．文字列 “abc”であれば，{“a”，“bc”}，{“ab”，“c”}，{“abc”}，{“a” ，“b”，“c”} の 4 通りのパターンが存在する．各パターンにおいて情報量を計算し，最も情報量が小さいものを文字列の持つ情報量とする．文字列 S の長さを $length(S)$ とすると，$P(t)$ は数式で $length(S) \leq 88k$ かつ $freq(t) > 1$ を満たす時，式 (4) のように計算される．

$$\hat{P}(t) = \frac{freq(t) - 1}{legth(L)} \tag{4}$$

ここで k はパターンの長さで，16 分音符に相当する長さを 1 とする．また L は判定対象の作曲者の楽曲群全体を文字列化した長さで，$freq(t)$ は，その文字列化した楽曲群のなかで，文字列 t が出現する頻度である．

実際の作曲者判定においては図 8 に示す通り，対象となる文字列化された楽曲に含まれる情報量を，作曲者ごとにグループ化された楽曲群を元に計算している．対象となる文字列を分割し，各パターンの生起確率は楽曲群から計測している．

文字列に含まれる全パターンについて情報量をもとめる場合，パターンごとに計算をおこなうと計算量が膨大になる．そこで，ダイナミックプログラミングを用いて情報量を削減する．文字列 “abc”について情報量を計算する場合を考える．文字が含まれていない，つまり 0 文字目の場合の情報量は 0 である．1 文字目 “a”が持つ情報量は $-\log_2 P(\text{“}a\text{”})$ となり，これは 0 文字目の情報量に 1 文字目の情報量を加算したものである．次に 2 文字目までの文字列 “ab”について考える．2 文字目までの文字列における分割は {“a”,“b”} と {“ab”} である．この時の各分割は式 (3) における π_k であり，各文字列が t である．文字列 “ab”の情報量 $I(\text{“}ab\text{”})$ は $-\log_2 P(\text{“}ab\text{”})$ もしくは $-\log_2 P(\text{“}a\text{”}) - \log_2 P(\text{“}b\text{”})$ であり，本手法ではより情報

量の小さい方を採用する．この時，$-\log_2 P(\text{“}a\text{”})$ は先に計算した 1 文字目の情報量と等しいため，すでに計算した $I(\text{“}a\text{”})$ と置き換えることができる．同様に，3 文字目までの文字列 “abc” における分割は {“a”，“bc”}，{“ab”，“c”}，{“abc”}，{“a” ，“b”，“c”} である．“a” は 1 文字目までの情報量 $I(\text{“}a\text{”})$ に，“ab” もしくは “a”，“b” は 2 文字目までの情報量 $I(\text{“}ab\text{”})$ に置き換えることができる．このように，各文字列における計算の一部をすでに計算した情報量と置き換えることで計算量を削減できる．

本研究では，作曲者が持つ固有パターンの長さを推定することを目的としている．従って，楽曲の情報量を計算する際文字列の長さを制限する．これにより，特定の長さ以上のパターンは計算されなくなるため，その正解数からどの長さが作曲者の特徴であるかが推定できると考えられる．本研究では，楽曲を文字列化する際の時間間隔は 16 分音符を基準とし，その時間の鍵盤の状態を 88 文字で表している．従って，1 音分とは 16 分音符 1 つ分の区間，すなわち 88 文字を表す．

4. 実験

4.1 評価方法

作曲者の判定は 5 人の作曲者（Bach, Chopin, Debussy, Mozart, Satie）のピアノ曲各 15 曲ずつ，計 75 曲に対して行う．これらの楽曲は先行研究 [5] と同じものであり，曲名はそちらを参照されたい．ある 1 曲を未知として判定をする時，5 つの楽曲群を用いて情報量を計算し，情報量が最も小さいグループの作曲者を未知の楽曲の作曲者とする．これを全ての楽曲に対して行い，判定結果の作曲者が合致していれば正解とする．

作曲者を判定する際，作曲者のグループに判定対象となるデータが存在すれば正解する確率が高くなってしまう．従って図 8 を図 9 のように変更し判定する．a_1,...,a_{15} は作曲者 A から E の楽曲を表している．図 9 は作曲者 A の楽曲 a_1 が判定対象の場合である．判定対象となる楽曲は作曲者 A のものであるため，作曲者 A の楽曲群から a_1 の楽曲を削除して判定する．同様の処理を全ての楽曲に対し行うことで作曲者を判定する．つまり，leave-one-out 交差検証による評価を行なっている．また本実験では，長さの制限を 1 音分，2 音分，4 音分，8 音分，16 音分，制限なしとして各々判定を行い，正解数を比較した．

4.2 実験結果

各遷移の長さについてそれぞれ判定を行なった結果を図 10 に示す．図 10 において正解数は全 75 曲のうち，正しく作曲者を判定できた曲数を表す．図 10 を見ると，2 音分と 4 音分の正解数の差は 1 曲である．このことから，作曲者を判定するには 2 音分の情報があれば十分であるという結果が得られた．

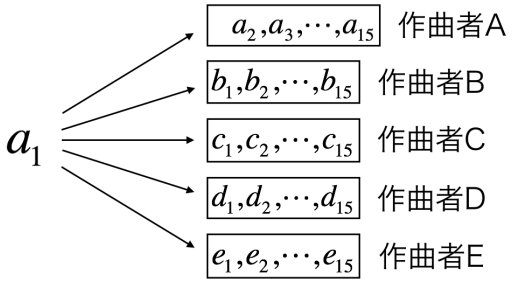

図 9 グループ化を用いた評価方法

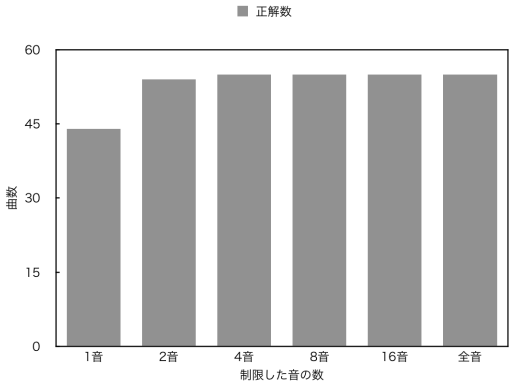

図 10 楽曲を文字列化した結果

5. 考察

本研究では作曲者の特徴が現れる長さについて，判定に用いる遷移の長さに制限をかけて作曲者を判定し検討した．その結果，正解数から作曲者の特徴として捉えられている長さとして 16 分音符 2 音分であることがわかった．実際にはより長いパターンを特定できればより有用であると考えていたが，そのような結果にはならなかった．しかしこの結果から，作曲者の特徴はある音から次の音へ遷移する時の，音の上がり下がりの情報が重要であると考えられる．また，一音の長さでも判定ができることから今後の課題として，作曲者判定の精度をさらに高めるために，文字列化の方法や情報量の計算方法を工夫し，より長い音の情報を得られるようにしたい．

6. おわりに

本研究では，情報量計算を用いた作曲者判定プログラムを用いて，作曲者の特徴が楽曲においてどの程度の長さに現れるかを分析する実験を行なった．実験では，計算する情報量の文字列長に制限をかけた時，その正解数から作曲者の特徴として捉えられていると考えられる遷移の長さについて分析を行った．その結果，作曲者の特徴として捉えられていると考えられる遷移の長さはおよそ 2 音であった．これは，隣り合う音の上がり下がりといった遷移情報が有用であるためと考えられる．また，和音を構成する音の組み合わせも作曲者を示す情報であると考えらる．

参考文献

[1] K. Adachi, M. Okabe, K. Umemura.: *Considering chord inversion on estimating composers of score*, Technical Report 2013-MUS-101, no. 5, Information Processing Society of Japan SIG Technical Report,(2013).
[2] R. B. Dannenberg, B. Thom, and D.Watson.: *A machine learning approach to musical style recognition*, in Proceedings of International Computer music Conference, pp. 344 - 347,(1997).
[3] S. K. Sawada, T.: *Composer classification based on patterns of short note sequences*, in Proceedings of the AAAI-2000 Workshop on AI and Music, pp. 24 - 27,(2000).
[4] 圧縮類似度における楽譜からの作曲者の判定, Data Engineering and Information Management（2013）
[5] A.Takamoto, M. Umemura, M.Yoshida, K.Umemura.: *Improving compression based dissimilarity measure for music score analysis*, in Advanced Informatics: Concepts, Theory And Application(ICAICTA), pp. 1 - 5,(2016).
[6] 中川聖一：情報理論の基礎と応用，近代科学社 (1992).