既存楽曲の特徴分析に基づく作曲支援に関する研究

久原 聖志† 牛尼 剛聡‡

†九州大学大学院芸術工学府 〒815-8540 福岡県福岡市南区塩原 4-9-1
‡九州大学大学院芸術工学研究院 〒815-8540 福岡県福岡市南区塩原 4-9-1
E-mail: †kuhara.s.000@s.kyushu-u.ac.jp, ‡ushiama@design.kyushu-u.ac.jp

あらまし 本研究は，既存の楽曲のデータを用いて楽曲の特徴を分析し，そこで得られた特徴に基づいて，ユーザの音楽制作を支援する手法を開発することを目的とする．本研究では，インターネット上に投稿された楽曲を解析していられたデータを対象に，楽曲の特徴を取得し，マルコフ過程としてモデル化することにより楽曲の生成モデルを構築する．そして，ユーザが提示した楽曲メロディの一部分に対して，その部分の続きに適した音の候補の提示を行い，補完を支援する．

キーワード 楽曲分析，機械学習，作曲支援

1. はじめに

近年，インターネットが普及し，YouTube やニコニコ動画等の動画共有サイト等のコンテンツ投稿サイトが一般的に利用されるようになった．動画共有サイトでは，専門家ではなくても，一般の人々が作品を投稿することで，他者に投稿した作品を鑑賞してもらう場が増加した．しかし，専門的な知識がない人々が簡単に作品を投稿できるため，すべての作品のクオリティが高いというわけではない．こうした中で，作曲に関して十分な知識と能力がないユーザが，少しでもクオリティの高い作品を制作できるように支援する機構に対する要求が存在する．

作曲を行う場合には，途中まで作成したメロディに続く適切なメロディが思い浮かばない場合や，作成したメロディに対して違和感を感じることが少なからず存在し，作曲活動が停滞する場合がある．

本研究では，既存の楽曲のメロディ，コード，楽曲構造を解析することにより得られた特徴に基づいて，専門的知識を持たない利用者による作曲活動を支援する手法を提案する．具体的には，ユーザが作成したメロディの一部分を指定すると，そのメロディの後にふさわしいメロディの候補を推薦する手法を提案する．

2. 関連研究

楽曲の生成に関しては，自動作曲を行う Orpheus[1]という研究が存在し，さらに歌詞の韻律を制約条件として最尤経路探索問題を解くことで任意の日本語の歌詞を用いた歌唱曲の生成を実現している．

また，対話型遺伝的プログラミングを行うことにより，クラシック音楽の作曲家が馴染みやすい遺伝子表現や作曲過程を特徴とした作曲支援システム[2]の構築や，学習したユーザの感性モデルから遺伝的アルゴリズムを通して自動作曲を行うといった研究[3]など，遺伝的な技術を用いた研究も存在する．これらは音楽理論や既存楽曲の特徴など，音楽的知識をシステムに実装している．しかし，自動作曲は，表現を自ら行いたいユーザの要求には答えられないと考えられる．さらに，自動作曲においては，必ずしも創作者が意図しないようなメロディが生成されることがあるという指摘もあり,ユーザがメロディの大まかな形のイメージ，つまり旋律包絡を持っていると考え[4]，与えられたメロディをフーリエ変換することにより旋律の編集を容易にすることで支援を実現するという研究[5]も行われている．しかし，これらの研究には音楽的な知識が利用されていない．

本研究では，既存楽曲の特徴を学習し，ユーザのメロディ入力に基づいて，ユーザに提示する音を決定する．また本研究では，既存楽曲を分析するにあたり，メロディを文章として捉え，言語処理における確率計算を用いてユーザに提示するメロディを計算する．また，自動作曲とは違いユーザのメロディ，あるいはユーザが持つメロディのイメージも利用し，候補の提示を行うシステムである．

3. 提案手法

3.1. 楽曲の分析

3.1.1. Songle

本研究では，多数の既存楽曲の特徴にもとづいてメロディの候補の提示を行う手法を開発することを目的とする．そのために，既存の楽曲の特徴を抽出する必要があるが，人間の手作業で膨大な楽曲と特徴を抽出することは困難である．そこで本研究では，既存の楽曲の分析のために Songle[6]を利用する．Songle は，動画サイトに投稿された楽曲の動画から，メロディ，コード，小節，拍子，サビといった楽曲構造を自動的に解析する Web サービスである．Songle では，コンピュータによる解析では間違いが含まれる可能性があるた

め,解析されたデータを人間が訂正することができる.

3.1.2. 楽曲データの取得とデータベースへの登録

Songle を利用した楽曲の解析結果は，Songle Widget API によって取得することができる．今回はメロディとコードを取得する.

Songle Widget API で取得可能なデータは,開始時間,内容（メロディであれば音高（ピッチ）を示す数字,コードであればコード名),インデックス,時間の長さである．本研究では，これらの中で，音の開始時間,内容，インデックスを利用する.

取得したデータは，メロディを構成する音高をインデックス順に並べ，一つ音高を一文字として，一文章化する.そのようにして文章化された文字列に対して,N-gram を用いて文字列を分割する．N-gram を用いた理由として，N が 2 以上であれば音高の遷移を一つのデータに登録することができ，またそのようなデータが大量に取得されたとき，そこから遷移確率を計算することができるからである.

N 個からなる音高の遷移によって構成されたタプルに対し，そこに含まれる音高の遷移が，どのコードに属しているかを調べ，そのタプルに格納する.

例えば，2-gram を対象とするとき，“ADFCBE”という列があり，ADF はコード 1 に，CBE はコード 2 に属しているとする. 文字列に含まれる 2-gram は, AD, DF, FC, CB, BE である．例えば DF について，これは D と F どちらとも同一のコード 1 に属しているため,

（D,F,コード 1）

という組が生成される．一方，FC の場合，F はコード 1 に, C はコード 2 にと, 属するコードが異なるため,F がコードにおいて末端であることを示すために，組の 2 番目には末端を意味する記号を格納する．これを［END］とする. また, C はコードの先端であるため,組の 1 番目には先端を意味する記号を格納する．これを［START］とする. これらをそれぞれに適用すると,

（F,[END],コード 1）

（[START],C,コード 2）

という組がそれぞれ生成される．これらをデータベースに登録する.

なお，”ADF”と”CBE”がそれぞれ違うコードに属していると判定するために，メロディおよびコードのデータに格納された開始時間を分析するという方法を取る.

3.2. モデルの生成

調性音楽には，音階と呼ばれる楽曲に利用できるピッチの系列が存在する[7]. コードとメロディの関係は,コードを構成する音高とメロディの音高について，すべてに対して同じ数を加減すれば，聞こえ方は違ったとしても同じ曲として聞こえる.

例えば，コード C 上で E,G,C というメロディが鳴っているとすると, コードの C, そしてメロディの E,G,C に対して半音を 2 回加える，つまり全音を 1 回加えるとすると，コードは D，メロディは F#,A,D となる．これらの 2 種類を聞くとき，同じ楽曲を聞いているように感じ，2 種類における違いは音階のみである.

3.1.2.においては, これらのことを考慮せずに絶対的な音高とコードを登録していた．しかし，これを他の音階でも用いるためには，絶対的なものとして登録されたメロディとコードの関係の差分を取り，相対的な特徴に変換する必要がある.

例えば 3.1.2.に示した手法で,

（E,C,Am）

という遷移データを登録したとする．これは，E→C という音の遷移が Am（イ短調）というコードの上で行われたことを意味する．これは，イ短調における絶対的なデータである．これを相対的な遷移データに変換する手法を次に示す.

まず，コードの根音の基準を C とする．C を 0 番目とすると，A は 9 番目となる．Am というコードを Cm に置き換えるとき，A から C に置き換わる際には 9 回半音が下がっている．これに従い，メロディの E,C にも 9 回半音を下げるという処理をしたとき，E,C はそれぞれ G,E♭（E♭は E の半音下がった音である）となる．コードの根音の基準を C と置くと決めたことにより，データベースに登録する際に必要なデータは G と E♭，そして変換前にコードの根音の次についていた単調を意味する「m」の 3 つである．つまり，遷移データは

（G,E♭,m）

という形になる．この処理を複数のデータについて行ったとき，同じコードの種類における音の遷移に関する遷移データが集まり，これが一つのモデルとなる.

例えば，同じコードの種類の遷移データを以下のように並べた.

（D,G,Gmaj），（E,F,Fmaj），（B,A,Amaj）

これらを先に示した方法で相対的な遷移データに変換すると,

（G,C,maj），（B,C,maj），（D,C,maj）

となる.（maj はメジャーという種類を意味する.）これらの遷移データからは，メジャーのコード上において，C という 0 番目の音高に対し，G という 7 番目の音高，B という 11 番目の音高，D という 2 番目の音高から遷移することがあるという知識が得られる.

このように絶対的なデータから相対的なデータに変換し，記録する処理を通して学習を行うことをモデルの生成と称している.

3.3. 候補の提示

3.2.において絶対的な遷移データからの変換により生成されたモデルから，同一コード内における遷移前の音と遷移後の音をひとまとめにし，得られた遷移データの中からその組み合わせを数え，遷移確率を計算した．3.1.2.における[START]と[END]をそれぞれ S と E とおき，12 の音高を組み合わせた音高の集合 $Note$ は，

$$Note = \{S,0,1,\ldots\ldots,10,11,E\}$$

と表される．$Note$ における数字と，相対的な音高との変換表を以下に示す．

表 1：変換表

数字	音高名
S	[START]
0	C
1	D♭
2	D
3	E♭
4	E
5	F
6	G♭
7	G
8	A♭
9	A
10	B♭
11	B
E	[END]

音nから音mまでの遷移の件数を$count_{n\to m}$と置くと，nから 14 種類ある遷移先のうちの m に遷移する確率は

$$p_{n\to m} = \frac{count_{n\to m}}{\sum_{k\in Note}(count_{n\to k})}$$

と表される．遷移元のnについて，集合 $Note$ にある音高全てへの遷移確率をまとめた 14 次ベクトルを$\boldsymbol{p_n}$とおくと，

$$\boldsymbol{p_n} = \begin{pmatrix} p_{n\to S} \\ p_{n\to 0} \\ \cdots \\ \cdots \\ p_{n\to 11} \\ p_{n\to E} \end{pmatrix}$$

と表される．また，このベクトルを $Note$ ごとに収集して一つの 14 次正方行列を作成する．その行列をP_{Chord}とおくと，

$$P_{\mathrm{Chord}} = (\boldsymbol{p_S} \quad \boldsymbol{p_0} \quad \cdots \quad \boldsymbol{p_{11}} \quad \boldsymbol{p_E})$$

$$= \begin{pmatrix} p_{S\to S} & p_{0\to S} & \cdots\cdots & p_{11\to S} & p_{E\to S} \\ p_{S\to 0} & p_{0\to 0} & \cdots\cdots & p_{11\to 0} & p_{E\to 0} \\ \cdots & \cdots & \cdots\cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots\cdots & \cdots & \cdots \\ p_{S\to 11} & p_{0\to 11} & \cdots\cdots & p_{11\to 11} & p_{E\to 11} \\ p_{S\to E} & p_{0\to E} & \cdots\cdots & p_{11\to E} & p_{E\to E} \end{pmatrix}$$

となる．この行列に，遷移前の時刻 t における音高に該当する成分を 1，それ以外を 0 とした 14 次ベクトルを$\boldsymbol{x_t}$とおく．P_{Chord}と$\boldsymbol{x_t}$をかけたとき，計算結果を$\boldsymbol{x_{t+1}}$とすることができ，この各成分は時刻 $t+1$ において，成分に紐付けられた音高に遷移する確率である．

例えばコードの種類 maj において，遷移前の音高が B であったとき，B に該当するのは 13 番目の成分であるので，

$$\boldsymbol{x_t} = \begin{pmatrix} 0 \\ 0 \\ \cdots \\ \cdots \\ 1 \\ 0 \end{pmatrix}$$

と表され，$\boldsymbol{x_{t+1}}$は，

$$\boldsymbol{x_{t+1}} = P_{maj}\,\boldsymbol{x_t}$$

という式で計算され，例えば時刻 t において音高が B であったときに時刻 $t+1$ において C である確率は，$\boldsymbol{x_{t+1}}$ の 2 番目の成分となる．

S 遷移前の音をn_t，遷移後の音をn_{t+1}，そしてn_tとn_{t+1}が属するコードを$chord$とする．上記のように計算された確率$P(n_{t+1}, chord|n_t, chord)$を推薦度として利用し，その推薦度を作曲者に提示することにより，作曲者にメロディのアイデアを与える．

4. 作曲支援インタフェース

次に，取得されたデータに基づいて，ユーザの作曲支援を行うインタフェースについて述べる．次に弾くべきノートを視覚的にユーザに提示するために，鍵盤型のインタフェースを開発した．これは，任意の鍵盤を押したときに，その押した音からの推薦度が高い遷移先の鍵盤が赤く表示され，推薦度の順位が下がるほど薄くなっていく．(ただしその押した鍵盤の音自身への遷移については，今回は考慮していない．) 例えば表 2，表 3 のデータをそのまま用いる場合，コードは Cmaj となる．ここで，例えばソの鍵盤を押したとき，ソは数字に変えると 7 であるので 7 からの遷移先が上位の順から赤く表示される．表 3 を参照すると，自身（7）を除いて上位から順に 4 つ挙げると 0，6，4，8 の推薦度が高い．今回インタフェースの様子を図 1 に示す．この図 1 から，0，4，6，8 に相当する部分ほど赤くなっているのがわかる．このようにして赤い部分を次に押すと，そこからまた推薦度に従って鍵盤が赤くなる．これに従って弾いていき，コード Cmaj において適切なメロディとなることを想定しているが，現段階ではまだ評価が不十分なため，今後評価を行っていく．

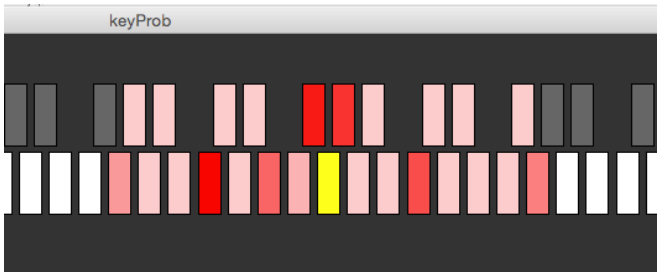

図 1 ：実装したインタフェース

5. 実験

2-gram で楽曲のデータを取得するプロトタイプシステムを実装し，Songle 上の「最近訂正された楽曲」に掲載されていた楽曲を取得した.「最近訂正された楽曲」を取得する楽曲に選んだ理由は，すでに人の手によって完全ではなくても修正が施されており，コンピュータによって解析されて以降，誰の手によっても修正されていない楽曲と比較して情報が正確であると考えたからである.

次に，その楽曲から，楽曲のサビの部分のデータを取得した．データベースに 3.1.2.で述べたように登録したのち，3.2.に述べたように相対的なデータに変換し，モデルを生成した．今回は，2,582 件分の楽曲についてのデータを取得し，990,439 件の遷移データからモデルを生成した．例として，コードの種類 maj におけるモデルから計算された遷移確率の表を以下に示す.

表 2：コードの種類 maj における
S,0,1,2,3,4,5 から各音高への遷移確率

	from S	from 0	from 1	from 2	from 3	from 4	from 5
to S	0.000	0.000	0.000	0.000	0.000	0.000	0.000
to 0	0.295	0.564	0.235	0.112	0.077	0.140	0.101
to 1	0.022	0.040	0.424	0.071	0.015	0.008	0.010
to 2	0.061	0.037	0.149	0.505	0.120	0.030	0.023
to 3	0.032	0.015	0.018	0.070	0.445	0.061	0.014
to 4	0.136	0.071	0.026	0.047	0.199	0.537	0.135
to 5	0.080	0.030	0.022	0.023	0.026	0.078	0.474
to 6	0.034	0.013	0.012	0.012	0.013	0.015	0.077
to 7	0.172	0.098	0.032	0.064	0.037	0.062	0.070
to 8	0.031	0.012	0.015	0.007	0.011	0.008	0.013
to 9	0.053	0.025	0.015	0.021	0.010	0.018	0.023
to 10	0.040	0.020	0.014	0.011	0.015	0.008	0.015
to 11	0.045	0.056	0.018	0.017	0.012	0.016	0.011
to E	0.000	0.018	0.021	0.039	0.022	0.021	0.035

表 3：コードの種類 maj における
6,7,8,9,10,11,E から各音高への遷移確率

	from 6	from 7	from 8	from 9	from 10	from 11	from E
to S	0.000	0.000	0.000	0.000	0.000	0.000	0.000
to 0	0.060	0.139	0.069	0.095	0.107	0.233	0.000
to 1	0.010	0.007	0.013	0.010	0.011	0.012	0.000
to 2	0.019	0.031	0.015	0.026	0.022	0.023	0.000
to 3	0.010	0.010	0.013	0.008	0.013	0.010	0.000
to 4	0.035	0.047	0.025	0.039	0.025	0.033	0.000
to 5	0.103	0.031	0.023	0.027	0.027	0.013	0.000
to 6	0.476	0.061	0.017	0.013	0.012	0.015	0.000
to 7	0.210	0.548	0.203	0.074	0.049	0.054	0.000
to 8	0.012	0.046	0.434	0.081	0.019	0.008	0.000
to 9	0.015	0.028	0.133	0.485	0.118	0.025	0.000
to 10	0.010	0.013	0.018	0.077	0.452	0.080	0.000
to 11	0.016	0.018	0.013	0.025	0.112	0.467	0.000
to E	0.025	0.023	0.024	0.043	0.031	0.027	0.000

表 2 と表 3 の結果から，コードの種類 maj において，[START]と[END]を除いてはどの遷移元からも，遷移元と同じ音高へと遷移している割合が一番高いことがわかった．同一コード内において，同じ音高をつづけて置く作曲が行われているケースが多いと考えられる.

6. まとめ

本研究では N-gram による文字列分割の技術を用いるという言語処理的なアプローチで既存楽曲の特徴の学習を試み，またその学習に基づいた確率の計算によりメロディの提示を行う手法を開発することを目的とする．今回は分析過程に音楽理論による制約を取り入れていないが，分析，あるいは提示の部分において音楽理論を応用した場合の提示精度についても検討したい.

今後は，ユーザによるメロディの入力，およびそのメロディ分析から候補の提示行うシステムの開発を行う.

またコードに関しては，一つの音を基本としたコードでもメジャーかマイナーの 2 通りに分かれ，またその中でもまた数種類に分かれるため，コードに関する集合についても考える必要がある.

評価に関しては，被験者実験を行う．創作活動として作曲をしている人を被験者として集め，被験者によって提示されたメロディを分析した結果提示された音に対して，よいか悪いかを数段階で評価してもらうという案と，作曲に関してのスキルは被験者に問わずに，提示するメロディは筆者自身で作成し，そこで提示される音に対して，被験者に評価してもらうという評価方法を考案している.

参　考　文　献

[1] 深山覚，中妻啓，米林裕一郎，酒向慎司，西本卓也，小野順貴，嵯峨山茂樹，”Orpheus: 歌詞の韻

律に基づいた自動作曲システム”，情報処理学会研究報告 2008-MUS-76

[2] 安藤大地，Dahlstedt Palle，Nordahl Mats，伊庭斉志，"対話型 GP を用いたクラシック音楽のための作曲支援システム"，芸術科学会論文誌，Vol.4(2005) No.2 pp.77-87

[3] 西川敬之，大谷紀子，福井健一，森山甲一，栗原聡，沼尾正行，” 楽曲の部分構造と全体構造を考慮した自動作曲システム”， 人工知能学会全国大会（第 23 回）論文集，1F1-1，2009.6

[4] 土屋裕一，北原鉄朗，”旋律包絡抽出に基づく直感的な旋律編集手法”，情報処理学会研究報告，Vol.2012-MUS-95-No.9，pp.41-46 (2012)

[5] 土屋裕一，北原鉄朗，”旋律概形を用いた旋律編集：概形レベルと音符レベルの編集をシームレスに行えるインターフェース”，情報処理学会インタラクション 2013

[6] 後藤真孝，吉井和佳，藤原弘将，Matthias Mauch，中野倫靖，”Songle : 音楽音響信号理解技術とユーザによる誤り訂正に基づく能動的音楽鑑賞サービス”，情報処理学会論文誌 Vol.54 No.4 1363-1372 (Apr,2013)

[7] 岩宮眞一郎，”図解入門　最新　音楽の科学がよく分かる本”，秀和システム，2012