

# N-Gram モデルを利用したマルチエージェントによる音楽の創発

## Multi-Agent Based Music Emergence by N-Gram Model

平山すみれ 成瀬美帆 向直人  
Sumire Hirayama Miho Naruse Naoto Mukai

### 1. はじめに

現在、コンピュータによって音楽制作を補助するシステムが広く普及している。その中でもコンピュータを利用した“自動作曲”に注目が集まっている。自動作曲とは、プログラムを介することで偶然性を持った曲を制作し、作曲過程の簡略化を図ることを指している。古くはモーツァルトの「音楽のサイコロ遊び」が知られる。サイコロを振って確率的に音楽を組み合わせ、偶然性の高い楽曲を制作するゲームである。

近年の研究では、コンピュータを利用することで、様々なリズムやメロディを実現する自動作曲システムが提案されている。例えば、岩井らは、画像の色彩情報を基に新しい曲を構成する MUSCAT というシステムを開発した[1]。色彩情報である色相、彩度、明度に関連付けられたコード進行を利用して楽曲を出力する。また、長名らは「リズムの進行」や「音の遷移」を遺伝的アルゴリズムの適用度として用いて、学習した曲の特徴を反映した新しい曲を作曲する手法を提案した[2,3]。これらの研究において生成される曲は、事前に定めたテンプレートや適用度を計算するための関数に依存する傾向が強い。

我々の研究では、既存の曲を基に新しい曲を構築するプロセスは既存研究と同じであるが、マルチエージェントシステムの仕組みを利用することで、個性の異なる様々なエージェントの相互作用により、自由な発想で作曲することを目的とする。さらに、作曲過程における「リズムの進行」や「音の遷移」を可視化することで、視覚的な楽しさに加え、音楽を学ぶ子供たちの補助ツールとして応用することを視野に入れる。魚住氏らを中心とした人工生命プロジェクト「SjQ (<http://sjq.jp/>)」は、マルチエージェントを用いて音楽を生成する「Gismo」を発表している。図1がGismoのスクリーンショットである。我々の研究では、このGismoのエージェントの相互作用に発想を得て、音符を表すエージェントの振る舞いを実装することにした。また、音楽の再生には、「ドレミ」などのテキストを音符とみなして音楽を演奏することが可能なフリーソフトのテキスト音楽「サクラ (<http://oto.chu.jp/>)」を利用する。

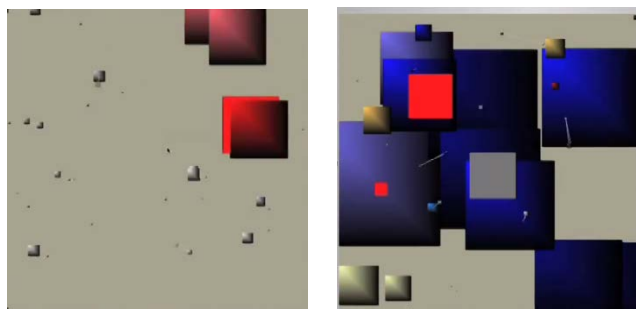


図1 Gismoのスクリーンショット  
(出展：[http://youtu.be/bAeH4gkOl\\_o](http://youtu.be/bAeH4gkOl_o))

本稿の構成は以下である。第2章では、テキスト音楽「サクラ」の音符の記述方法に関して述べる。第3章では、実装したエージェントによる作曲方法について説明する。第4章では作曲事例を紹介し、第5章で本稿をまとめる。

### 2. テキスト音楽「サクラ」

酒徳氏が制作した「サクラ」は、「ドレミ」などの文字列をエディタに記述すると、音符とみなし演奏してくれるテキスト音楽ソフトである。楽譜を記述するための文法はコンピュータ用の演奏情報の記述方式である MML (Music Macro Language) を基本としている。「サクラ」を利用して作成された楽曲は作者のウェブページ (<http://oto.chu.jp/>) で公開されている。「サクラ」では、「4分音符・8分音符」などの基本的な音符、元の音符の半分の長さをプラスした付点音符、シャープ (#)、フラット (b) などの臨時記号、決められた区間を繰り返し演奏するリピート (:)、音量の強弱など多彩な表現が可能であり、本来の楽譜に限りなく忠実に演奏することができる。

本研究では、「サクラ」の文法に従って記述された楽譜をマルチエージェント・シミュレータの「Artisoc」で読み込むことで、原曲の特徴を活かした新しい曲を生成する。「サクラ」では様々な表現が可能であるが、我々の開発したモデルでは、表1に示す音符表記のみを採用する。例えば、音の長さを変化させるには、ドレミなどの階名の前に「音符2」、「音符4」、「音符8」などの表記を付ける。また、音階を変化させるには、「'」や「”」を付けることで、1オクターブ上や下の音を演奏することができる。

表1 利用可能な音符表記

音符表記	意味
音符2ド	2分音符のド
音符4ド	4分音符のド
音符8ド	8分音符のド
'ド	1オクターブ上のド
”ド	1オクターブ下のド
ッ	休符

例えば、童謡の「ちょうちょう」を「サクラ」の文法で記述した楽譜が図2である。このように階名を記述するだけで、文字が音となり演奏される。

```
ソミミ フレレド ドレミフ ソソソッ  
ソミミ フレレド ミソソ ミミミッ  
レレレレ レミフッ ミミミミ ミフソッ  
ソミミ フレレド ミソソ ミミミッ
```

図2 ちょうちょうの楽譜

### 3. N-gram モデルを利用した作曲

「サクラ」の文法で記述した楽譜を「Artisoc」で読み込み、N-gram モデルに従って解析する。N-gram モデルは楽譜の解析手法として良く利用される。石田らは、N-gram モデルを利用して、即興演奏の不自然な旋律を自動的に修正するシステムを提案した[4]。N-gram モデルにより発生確率の低い外れた音が確率的に補正される。また、大槻らは N-gram モデルと隠れマルコフモデル (HMM) を利用して、音響信号から音符列を復元する手法を提案した[5]。

本稿では、N-gram モデルの解析結果に基づき、エージェントに音符列を割り当てる。図3が「Artisoc」のスクリーンショットである。背景画像として「音符の無料イラスト集 (<http://azukichi.net/melody.html>)」で提供されている画像を利用した。

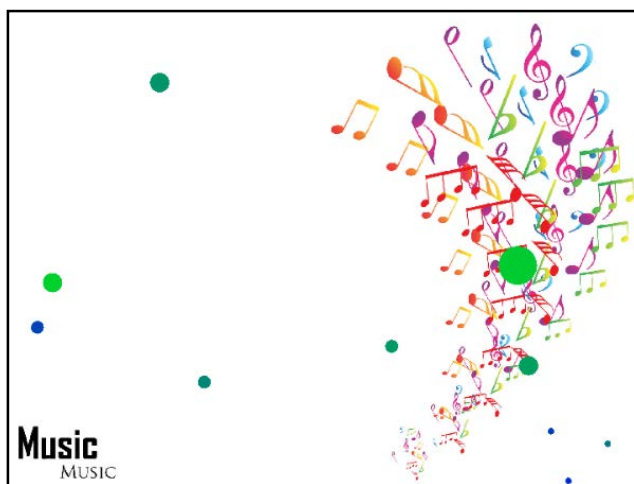


図3 Artisocのスクリーンショット

図中の複数の円が音符列を表すエージェントである。エージェントは一定の速度で直進し、空間の端に到達するとループして、空間内に再び出現することを繰り返す。エージェントは下記のいずれかの条件を満たすとき、対応する音を演奏する。エージェントの速度は固定であるため、他のエージェントとの衝突がなければ、一定のリズムで対応する音が演奏されることになる。また、エージェントの速度が速ければ、リズムも速くなり、エージェントの大きさが大きければ、衝突する確率が高くなる。

- エージェントが空間の端に到達したとき
- エージェント同士が衝突したとき

N-gram モデルとはテキストから切り出した一定数の文字の並びのことである。N-gram モデルには、1つの文字を表すユニグラム (uni-gram)、2つの連続する文字を表すバイグラム (bi-gram)、3つの連続する文字を表すトライグラム (tri-gram) がある。図2に示した「ちょうちょう」の楽譜を例に挙げると、ユニグラムであれば、図4に示すように、エージェントには「ミ」、「レ」、「ソ」などの音符が割り当てられる。バイグラムであれば、図5に示すように、「ミミ」、「レレ」、「ミフ」などが割り当てられ、トライグラムであれば、図6に示すように、「ミミミ」、「ソミミ」、「レレレ」などが割り当てられる。バイグラ

ムやトライグラムを表すエージェントは連続する音符が割り当てられるが、同時に複数音を演奏するのではなく、各ステップに1音ずつ演奏する。例えば、「ソミミ」であれば、最初に「ソ」、次のステップで「ミ」、さらに次のステップで「ミ」を演奏し、これを繰り返す。

ミ	頻度:23	倍率: 4.1	速度: 5.0
レ	頻度:13	倍率: 2.3	速度: 3.0
ソ	頻度:11	倍率: 2.0	速度: 2.0
フ	頻度:6	倍率: 1.1	速度: 2.0
ド	頻度:3	倍率: 0.5	速度: 1.0

図4 ユニグラム

ミミ	頻度:13	倍率: 3.0	速度: 4.0
レレ	頻度:8	倍率: 1.8	速度: 2.0
ミフ	頻度:5	倍率: 1.0	速度: 2.0
ソミ	頻度:5	倍率: 0.8	速度: 1.0
ソソ	頻度:4	倍率: 0.8	速度: 1.0

図5 バイグラム

ミミミ	頻度:7	倍率: 2.5	速度: 3.0
ソミミ	頻度:5	倍率: 1.3	速度: 2.0
レレレ	頻度:4	倍率: 1.3	速度: 2.0
フレレ	頻度:3	倍率: 1.3	速度: 2.0
ミミフ	頻度:3	倍率: 0.8	速度: 1.0

図6 トライグラム

複数のエージェントが同時に演奏の条件を満たす場合は、下記のいずれかの方法で処理する。エージェントが多く存在する場合、和音が高確率で発生し、聴きづらい音楽になってしまうことが多い。現状では、出現頻度の高い音符列を採用する方が、聴きやすい音楽になる傾向がある。また、「サクラ」では複数の音符表記を鍵括弧 (「」) で括弧することで和音を表現することができる。

- 和音 (同時に演奏) を生成する
- 出現頻度の高い音符列を優先する

各エージェントの大きさや速度は、割り当てられた音符列の出現頻度に応じて変化させる。まず、エージェントの大きさに関して定義する。割り当てられた音符列の出現頻度を  $f$ 、音符列の総数を  $F$ 、大きさの最大値を  $S$  とすると、各エージェントの大きさ  $s$  は下記の式で与える。

$$s = \frac{(S \times f)}{F} \quad (1)$$

例えば、「ちょうちょう」のユニグラムでは、大きさの最大値  $S = 10$  のとき、音符「ミ」のエージェントの大きさは  $(10 \times 23)/56 \cong 4.1$  となる。同様に音符「レ」のエージェントの大きさは  $(10 \times 13)/56 \cong 2.3$  となる。

次に、エージェントの速度に関して定義する。割り当てられた音符列の出現頻度を $f$ 、音符列の総数を $F$ 、速度の最大値を $V$ とすると、各エージェントの速度 $v$ は下記の式で与える。

$$v = \frac{(V \times f)}{F} + 1 \quad (2)$$

例えば、「ちょうちょ」のユニグラムでは、速度の最大値 $V = 10$ のとき、音符「ミ」のエージェントの速度は $(10 \times 23)/56 + 1 \cong 5$ となる。同様に音符「レ」のエージェントの速度は $(10 \times 13)/56 + 1 \cong 3$ となる。大きさや速度を出現頻度に応じて変化させることで、出現頻度が高い音符列を優先的に演奏し、個性ある旋律を生成する。

#### 4. 作曲事例

本章では、我々が構築した Artisoc モデルを用いた作曲事例を紹介する。原曲として、「ちょうちょう」「メリーさんの羊」「メヌエット」の3曲を用いた。モデル実行開始時におけるエージェントの初期位置や進行方向はランダムに決定される。このため、生成される曲も、実行する毎に異なった音符列になる。ここでは、生成された曲の中でも、より特徴的な曲をピックアップして紹介する。音符列の長さはモデルのステップ数に一致するが、ここでは20ステップまでの音符列を掲載する。また、複数のエージェントが同時に演奏の条件を満たした場合は、出現頻度の高い音符列を優先することとした。

##### 4.1 「ちょうちょう」を基にした作曲

童謡「ちょうちょう」を基に、ユニグラム、トライグラムで作曲した楽譜が図7、8である。図2に示したように、「ちょうちょう」の楽譜は全て4分音符で構成されている。ユニグラムの最頻値は「ミ」の23回、トライグラムの最頻値は「ミミミ」の7回である。楽譜中の最頻値は網掛で示される。生成されたいずれの楽譜においても「ミ」の出現頻度が圧倒的に高くなるのが分かる。原曲の雰囲気は残るものの、面白みの少ない音楽になってしまった。より良い音楽に近付けるには、パラメータの調整に加え、音の長さや音階を偶発的に変える必要があると思われる。

「音符 4 ソ」	「音符 4 レ」
「音符 4 ミ」	「音符 4 ミ」
「音符 4 ミ」	「音符 4 ミ」
「音符 4 ミ」	「音符 4 ミ」
「音符 4 レ」	「音符 4 ミ」
「音符 4 ソ」	「音符 4 フ」
「音符 4 ミ」	「音符 4 レ」
「音符 4 レ」	「音符 4 ミ」
「音符 4 ミ」	「音符 4 ソ」
「音符 4 フ」	「音符 4 ミ」

図7 ちょうちょう (ユニグラム)

「音符 4 ド」	「音符 4 ミ」	「音符 4 ソ」
「音符 4 ミ」	「音符 4 ミ」	「音符 4 ミ」
「音符 4 ミ」	「音符 4 ミ」	「音符 4 ミ」
「音符 4 レ」	「音符 4 レ」	「音符 4 レ」
「音符 4 ソ」	「音符 4 ミ」	「音符 4 ミ」
「音符 4 ミ」	「音符 4 ミ」	「音符 4 ミ」
「音符 4 ソ」	「音符 4 ミ」	

図8 ちょうちょう (トライグラム)

##### 4.2 「メリーさんの羊」を基にした作曲

童謡「メリーさんの羊」を基に、ユニグラム、トライグラムで作曲した。原曲の楽譜は図9、ユニグラムは図10、トライグラムは図11である。「メリーさんの羊」の楽譜は、4分音符だけでなく、2分音符と8分音符も用いられている。ユニグラムの最頻値は「音符 4 ミ」の8回、トライグラムの最頻値は「音符 4 ド 音符 4 レ 音符 4 ミ」などを含む6つの音符列が2回である。生成されたいずれの楽譜においても、音の長短が含まれているため、表現力の高い曲が生成された。また、ユニグラムに比べ、トライグラムは最頻値の音符列に「ミ」が多く含まれていたため、単調な音楽になってしまった。頻度の少ない音符列も一定の確率で楽譜に含めるなどの工夫が必要だと思われる。

音符 4 ミ.音符 8 レ.音符 4 ドレ
音符 4 ミミ.音符 2 ミ
音符 4 レレ.音符 2 レ
音符 4 ミソ.音符 2 ソ
音符 4 ミ.音符 8 レ.音符 4 ドレ
音符 4 ミミ.音符 2 ミ
音符 4 レレ.音符 8 レ
音符 2 ド

図9 メリーさんの羊 (オリジナル)

「音符 4 ミ」	「音符 4 ミ」
「音符 4 ミ」	「音符 4 ミ」
「音符 4 レ」	「音符 4 ド」
「音符 2 ド」	「音符 2 ド」
「音符 4 ソ」	「音符 4 ソ」
「音符 4 ミ」	「音符 4 ミ」
「音符 8 レ」	「音符 4 ソ」
「音符 4 ソ」	「音符 4 ミ」
「音符 4 レ」	「音符 4 ソ」
「音符 4 ソ」	「音符 4 ソ」

図10 メリーさんの羊 (ユニグラム)

「音符 2 ミ」	「音符 4 レ」	「音符 4 レ」
「音符 4 ミ」	「音符 4 ミ」	「音符 2 ミ」
「音符 4 ミ」	「音符 2 ミ」	「音符 4 レ」
「音符 4 ド」	「音符 4 レ」	「音符 4 ミ」
「音符 4 ミ」	「音符 4 ミ」	「音符 2 ミ」
「音符 4 ミ」	「音符 2 ミ」	「音符 4 レ」
「音符 4 レ」	「音符 4 ミ」	

図11 メリーさんの羊 (トライグラム)

### 4.3 「メヌエット」を基にした作曲

舞曲「メヌエット」を基に、ユニグラム、トライグラムで作曲した。原曲の楽譜は図12、ユニグラムは図13、トライグラムは図14である。「メヌエット」の楽譜は、音の長さの変化に加え、音階の変化も多く含まれている。ユニグラムの最頻値は「音符4ソ」の8回、トライグラムの最頻値は「音符8ド音符8シ音符8ラ」の4回である。いずれの音符列も出現頻度はそれほど高くないため、生成された楽譜には様々な音符列が含まれている。図15がトライグラムで作曲している様子である。エージェントが多く存在することから、音符列のバリエーションも多いことが分かる。また、どのエージェントも比較的小さいため、衝突する可能性は低く、一定のリズムを刻みやすい状態にあることが分かる。よって、他の2曲に比べ、より音楽的に複雑で面白い曲が生成された。特にトライグラムにおいては、最頻値の「音符8ド音符8シ音符8ラ」が定期的に含まれることで、心地よいリズムを生み出しているように感じた。

音符4レ音符8ソラシド  
 音符4レソソ  
 音符4ミ音符8ドレミフ#  
 音符4ソソソ  
 音符4ド音符8レドシラ  
 音符4シ音符8ドシラソ  
 音符4フ#音符8ソラシソ  
 音符4シ音符2ラ  
 音符4レ音符8ソラシド  
 音符4レソソ  
 音符4ミ音符8ドレミフ#  
 音符4ソソソ  
 音符4ド音符8レドシラ  
 音符4シ音符8ドシラソ  
 音符4ラ音符8シラソフ#  
 音符2ソ。

図12 メヌエット (オリジナル)

「音符8ラ」 「音符8ラ」  
 「音符8ド」 「音符4ソ」  
 「音符8シ」 「音符8ソ」  
 「音符8レ」 「音符4シ」  
 「音符8ラ」 「音符4レ」  
 「音符8ラ」 「音符8ラ」  
 「音符4ソ」 「音符8ソ」  
 「音符8シ」 「音符8シ」  
 「音符8ド」 「音符8シ」  
 「音符8ラ」 「音符8ソ」

図13 メヌエット (ユニグラム)

「音符8シ」 「音符8ラ」 「音符8ソ」  
 「音符8シ」 「音符8ラ」 「音符8ソ」  
 「音符8ド」 「音符8シ」 「音符8ラ」  
 「音符4シ」 「音符8ド」 「音符8シ」  
 「音符4ミ」 「音符8ド」 「音符8レ」  
 「音符8ド」 「音符8シ」 「音符8ラ」  
 「音符4ソ」 「音符4ミ」

図14 メヌエット (トライグラム)

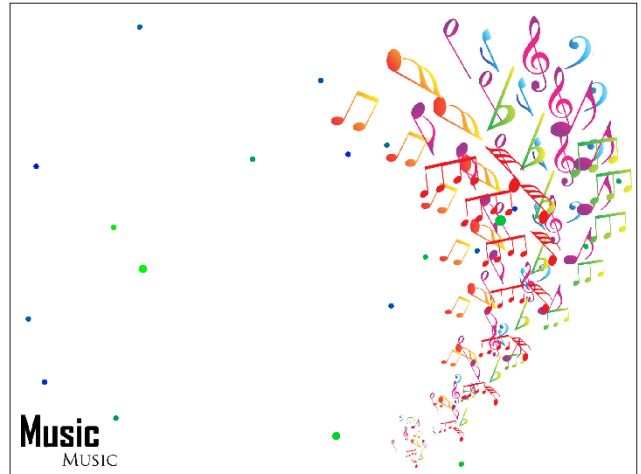


図15 Artisocのスクリーンショット (メヌエット)

## 5. まとめ

本研究では、近年注目されるコンピュータを利用した音楽制作をテーマに、マルチエージェントの概念を導入した自動作曲のための Artisoc モデルを制作した。N-gram モデルを利用して原曲に含まれる音符列をエージェントに割り当てる。このとき、割り当てられた音符列の出現頻度に応じてエージェントの速度や大きさが設定される。エージェントが特定の条件を満たすときに、割り当てられた音符列を演奏することで、原曲をアレンジした新しいリズムとメロディの音楽が生成される。また、エージェントの動きを視覚的に楽しむために、Artisoc モデルはポップな雰囲気デザインした。原曲として「ちょうちょう」「メリーさんの羊」「メヌエット」の3曲を採用し、ユニグラムとトライグラムで作曲を行なった。音の長さや音階の変化の少ない原曲の場合は、単調な音楽となってしまうことが分かった。一方、原曲に様々なパターンの音符列が含まれている場合は、生成される曲は複雑で音楽的に面白い音楽が生成されることが分かった。

今後の課題として、より自由度の高い曲を生成するためには、原曲を1つに絞るのではなく、複数の曲を利用することが考えられる。「ちょうちょう」などのシンプルな曲でも、他の曲と組み合わせることで、より音楽的に面白い曲に生まれ変わる可能性がある。また、本稿では、「自動作曲」をテーマとしたが、「自動歌詞」も興味深いテーマだと考えている。エージェントに、音符列に加え歌詞を割り当てることで、より発展的な音楽制作システムの構築が可能だと考えている。

### 謝辞

テキスト音楽「サクラ」の開発者である酒徳氏に深く感謝致します。

### 参考文献

- [1] 岩井憲一, "色彩情報に基づく自動作曲システム MUSCAT の作曲手法について", 映像情報メディア学会冬季大会講演予稿 (2013)
- [2] 高野美央, 長名優子, "N グラムモデルと遺伝的アルゴリズムを用いた複数の曲の構成を考慮した自動作曲システム", 情報処理学会第75回全国大会 (2013)
- [3] 菊地基希, 長名優子, "遺伝的アルゴリズムを用いたコード進行を考慮", 情報処理学会第76回全国大会 (2014)
- [4] 石田克久, 北原鉄朗, 武田正之, "N-gram による即興演奏の旋律補正", 情報処理学会論文誌, Vo.45, No.3, pp.743-746 (2004)
- [5] 大槻知史, 中井満, 下平博, 嵯峨山茂樹, "HMM と音符 n-gram を用いた音楽リズム認識", 情報処理学会研究報告, Vol.2001, No.103, pp.61-68 (2001)