Vol.2018-MUS-120 No.14
2018/8/22

コードとメロディの階層的生成モデルに基づくインタラクティブ作曲システム

津島 啓晃[1,a)]　中村 栄太[1,b)]　吉井 和佳[1,c)]

概要：本稿では，ユーザが初めにメロディを与えたのちに，メロディとコード進行に関して，一方に対してもう一方を自動生成する操作を繰り返すことによって，楽曲を洗練させることができるインタラクティブ作曲システムを提案する．従来の自動作曲システムでは，メロディやコード進行の完全な自動生成が目標とされているため，生成結果の修正は手動で行う必要があるという課題があった．また，従来のコードのマルコフ性を仮定したコード進行生成手法では，コード進行に内在する繰り返し構造が考慮されないという課題もあった．この問題を解決するため，コードとメロディに関する統一的な階層的生成モデルを定式化し，そのモデルを予め学習しておくことで，現在得られているメロディやコード進行の中から，ユーザが指定した一部のみを事後分布に従ってサンプリングする手法を提案する．また，被験者実験による主観評価によって提案システムにユーザの楽曲制作を支援するシステムとしての高い有用性があることが示せた．

1. はじめに

作曲は，音楽的知識を持つ人間によって行われる高度な制作行為であることから，音楽的知識を持たない人の楽曲制作を支援するために，計算機による自動作曲に関する研究がさかんに行われている．従来研究では，メロディあるいはコード進行の完全な自動生成を目的としていたが，実際に作曲が行われるときには，メロディとコード進行の一部を少しずつ編集することで，楽譜全体が音楽的にふさわしくなるように洗練していくという過程が存在する．我々の目的は，音楽的知識のないユーザが自身の好みを反映しながら楽曲を編集することを支援できるようなインタラクティブ作曲システムを構築することである．

また，コードやメロディに関する統一的な評価基準のもとで，コードとメロディを最適化できることも重要な課題として存在する．楽曲を少しずつ洗練するためには，メロディからコード進行を生成する和声付け手法 [1–4] と，コード進行からメロディを生成する手法 [5,6] を繰り返して用いるという方法が考えられるが，その手法では，２つの手法で別々の評価基準のもと生成しているため，システムが楽曲全体の最適性を担保することができない．

楽曲の雰囲気は，コード進行とメロディによって特徴づけられるため，楽曲を制作する際には，コードとメロディにおける複雑な構造や依存性を考慮しないといけない．音楽的にふさわしいコード進行を生成するためには，３種類のコード記号のカテゴリ (tonic (T)，dominant (D)，subdominant (SD)) によって構成されるコードの機能和声という概念が考慮される．これは，自然言語における品詞の概念に一致する．さらに，コード記号の機能和声で表される系列は木構造によって記述されることが知られている [7,8]．したがって，ユーザの楽曲制作の支援を計算機的な方法で支援するときにも，このようなコード進行における階層的な木構造を考慮することが望ましいと考えられる．

以上をふまえ，本稿では，音楽的知識を持たないユーザが楽曲（メロディとコード進行）を編集することを支援するインタラクティブ作曲システムを提案する．ユーザは楽曲の編集操作として，(1) 現在のメロディ全体に対して，コード記号列とその背後にある木構造を同時に最適化する，

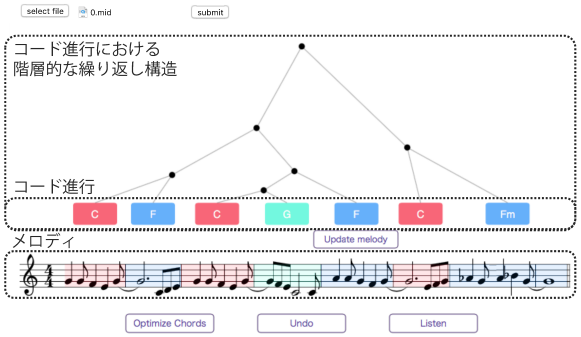

図 1: 本稿で提案するインタラクティブ作曲システムのユーザインタフェース

1 京都大学
a) tsushima@sap.ist.i.kyoto-u.ac.jp
b) enakamura@sap.ist.i.kyoto-u.ac.jp
c) yoshii@kuis.kyoto-u.ac.jp

(2) 選択したコードのオンセット位置を最適化する，(3) 隣り合ったコードを単一のコードにマージする，または単一のコードを 2 つのコードに分割する，(4) 選択したコードの区間にあるメロディを変更する，といった 4 種類の操作を選択することができる．

また，コード進行とメロディに関して統一的な評価基準を構築するために，我々は，(1) 確率的文脈自由文法 (PCFG) に基づくコード記号列生成モデル，(2) マルコフモデルに基づくコードまたはメロディのリズム生成モデル，(3) コードの条件付きマルコフモデルに基づくメロディの音高遷移モデルからなる階層的生成モデルを提案する (図 2)．PCFG は，その非終端記号とルール確率によって，コード記号の構文的役割とその繰り返し構造が捉えられることを期待して用いる．また，マルコフモデルに基づくメロディの音高遷移モデルを改良するため，より長期的なメロディの文脈を捉えることのできる長短期記憶 (LSTM) ネットワークを用いる．さらに，以上の生成モデルをあらかじめ学習しておくことによって，コード進行とメロディに関して，ユーザが気に入らない箇所を条件付き事後分布にしたがって更新する手法を提案する．

2. 関連研究

本章ではコード進行の自動生成とメロディの自動生成に関する関連研究をそれぞれ紹介する．

2.1 コード進行の自動生成

メロディに対するコード進行の自動生成（自動和声付け）に関する研究は，本研究を含むコード記号列を生成する研究と，四声合唱など複数声部を生成する研究に分けられる．前者の方向性では，Simon ら [2] は，コード記号の遷移をマルコフモデルによって表現した HMM に基づく自動和声付けシステム *MySong* を提案した．Raczyński ら [9] は，メロディと時変的な調情報に条件付けられたマルコフモデルに基づく統計的な手法を提案した．Tsushima ら [3] は，PCFG によって表現されるコード記号列の階層的な繰り返し構造と，マルコフモデルによって表現されるメロディの音高遷移を考慮した和声付け手法を提案した．De Prisco ら [4] は，ベースラインのみを入力として，ベースラインの遷移とコード進行の間にある依存性を表現したニューラルネットワークに基づく和声付け手法を提案した．

後者の方向性では，Ebcioğlu [10] は，バッハの四声合唱をルールベースによって生成する手法を提案した．また，遺伝子アルゴリズムを用いた研究 [11–13] も多く行われている．Allan ら [14] は，コードを隠れ状態，音符を出力状態として表現した HMM に基づく手法を提案した．また，コードの継続長を明示的に表現するために隠れセミマルコフモデル (HSMM) に基づく手法も提案された [15]．Paiement ら [16] は，コード記号表記を分割し，それぞれ

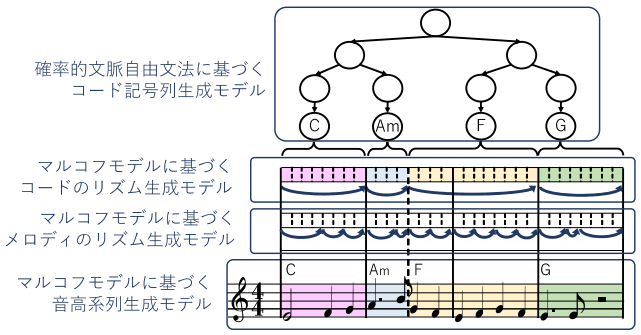

図 2: コード記号，コードのリズム，メロディに関する確率的生成モデル

に対応する階層的な時間スケールを考慮することによって，コード進行を表現する木構造モデルを提案した．また，より自然な四声部パートを生成するために，リカレントニューラルネットワーク（RNN）を用いた手法も提案されている [17]．

2.2 メロディの自動生成

メロディの自動生成に関しても多くの研究がなされている [5, 6, 18–21]．Fukayama ら [5] は，歌詞を入力として，歌詞の韻律とメロディの遷移をもとにメロディを生成することのできる自動作曲システム Orpheus を提案した．Roig ら [6] は，メロディのリズムのパターンや音高の概形に関する確率モデルを用いて単旋律のメロディを生成する手法を提案した．

近年では，深層学習の手法を適用した研究も多く存在する．例えば，Google Magenta Project [18] では，楽曲の長期的な依存性を反映するために，RNN が用いられている．Yang ら [19] は，より実際の音楽に近いメロディを生成するために，敵対的生成ネットワーク（GAN）と畳み込みニューラルネットワーク（CNN）を組み合わせた手法を提案した．Mogren ら [20] は，メロディ生成のために連続系列データに対して RNN に基づいて敵対的学習を行う手法を提案した．一方で，メロディの時間的な依存性を表現する RNN に条件付けられた制限ボルツマンマシン（RBM）を用いて多重音の旋律を生成する手法も提案されている [21]．さらに，Eck ら [22] によって，メロディの音符間の遷移やメロディとコード記号間の相互依存性を捉える特徴量に基づく LSTM に基づいて，メロディとコード進行の両方を生成する手法も提案されている．

3. 提案システム

提案システムでは，ユーザによってコード進行とメロディを少しずつ更新するための操作が，Web API 上で行えるように実装されている (図 1)．システムを使うにあたって，ユーザは最初に 8 小節分のメロディのデータをアップロードし，自動的にそのメロディに対してふさわしいコード進行が生成される．このとき，コードの開始位置は小節

線とする．そして，以下のアレンジ操作を行うことで楽曲が洗練できる．

- **コード記号列の更新**: 現在のメロディに対してコード進行とその背後にある木構造を同時に最適化する．
- **コードの開始位置の更新**: ユーザが選択したコードの開始位置（1 箇所）を最適化する．
- **コードの分割**: ユーザが選択した 1 つのコードを 2 つのコードに分割する．
- **コードの統合**: ユーザが選択した隣り合った 2 つのコードを 1 つに統合する．
- **メロディの更新**: ユーザが選択したコードの区間中のメロディ（の一部）を更新する．

4. 確率モデル

本章では，コード進行とメロディの階層的な生成過程を表現する統一的な生成モデルの定式化と学習方法について述べる．提案モデルは，4 つのサブモデルによって構成され，それぞれ独立に事前学習を行う．

4.1 数学的表記法

まず，コードとメロディの開始位置は，1 6 分音符が最小単位であると仮定する．L を入力する楽曲の小節数（本稿では $L = 8$）と記し，$T = 16L$ を楽曲中の 16 分音符の総数と記す．コード記号系列は，$\boldsymbol{z} = \{z_n\}_{n=1}^{N}$，コードの開始位置の系列は，$\boldsymbol{\phi} = \{\phi_n\}_{n=1}^{N}$ と記す．ここで，N は楽曲中のコードの数であり，ϕ_n は，0 から $T-1$ の整数値をとりうる．同様に，コード z_n の区間にあるメロディの音高及び開始位置を $\boldsymbol{x}_n = \{x_{n,i}\}_{i=1}^{I_n}$，$\boldsymbol{\psi}_n = \{\psi_{n,i}\}_{i=1}^{I_n}$ とそれぞれ表す．ここで，I_n は，その区間に存在する音符の数とする．また，$x_{n,i}$ は，32 から 93 までの MIDI ノートナンバーの値をとり，$\psi_{n,i}$ は，ϕ_n から $\phi_{n+1} - 1$ までの値をとる．メロディ全体は，$\boldsymbol{x} = \{\boldsymbol{x}_n\}_{n=1}^{N}$ 及び $\boldsymbol{\psi} = \{\boldsymbol{\psi}_n\}_{n=1}^{N}$ によって表され，$I = \sum_{n=1}^{N} I_n$ は，全体の音符数を表す．

また，PCFG にしたがってコード記号列 $\boldsymbol{z}$ を導出する木を $\boldsymbol{t}$ と記し，$t_{m:n}$ を，コード部分列 $\boldsymbol{z}_{m:n} = \{z_m, ..., z_n\}$ を導出する内側の部分木とする．本稿では簡単のため，$t_{m:n}$ によってその部分木の頂点ノードも表す．また，$t_{\neg m:n}$ を $z_{1:m-1}$, $t_{m:n}$, and $z_{n+1:N}$ を導出する外側の部分木であるとする．

4.2 階層モデルの定式化

我々は，コード記号列 $\boldsymbol{z}$ とその背後の木構造 $\boldsymbol{t}$，コードの開始位置 $\boldsymbol{\phi}$，メロディの音高 $\boldsymbol{p}$，開始位置 $\boldsymbol{\psi}$ に関する生成過程を統一的な確率モデルで定式化する．

コード記号列 $\boldsymbol{z}$ 及びその導出木 $\boldsymbol{t}$ は，確率的文脈自由文法 $G = (V, \Sigma, R, S)$ によって生成されると仮定する．ここで，V は非終端記号の集合，Σ は終端記号 (コード記号) の集合，R はルール確率の集合，S は開始記号 (構文木の根ノードにあたる非終端記号) とする．ルール確率は次の 3 種からなる．$\theta_{A\to BC}$ は非終端記号 $A\ (\in V)$ が 2 つの非終端記号 $B, C\ (\in V)$ に分岐する確率であり，$\eta_{A\to\alpha}$ は $A\ (\in V)$ が終端記号 $\alpha\ (\in \Sigma)$ を出力する確率である．λ_A は非終端記号 A が終端記号を出力する確率である．これらの確率については，以下の等式が成り立つ．

$$\sum_{B,C\in V} \theta_{A\to BC} = 1 \quad \sum_{\alpha\in\Sigma} \eta_{A\to\alpha} = 1 \tag{1}$$

また，$\boldsymbol{\theta}_A = \{\theta_{A\to BC}\}_{B,C\in V}$，$\boldsymbol{\eta}_A = \{\eta_{A\to\alpha}\}_{\alpha\in\Sigma}$，$\boldsymbol{\theta} = \{\boldsymbol{\theta}_A\}_{A\in V}$，$\boldsymbol{\eta} = \{\boldsymbol{\eta}_A\}_{A\in V}$，$\boldsymbol{\lambda} = \{\lambda_A\}_{A\in V}$ を定義する．同様の表記法を本稿を通して用いる．

コードの開始位置 $\boldsymbol{\phi}$ に関するマルコフモデルは，コードの 16 分音符単位での開始位置 (小節内の相対的な拍位置) に関する以下の遷移確率によって記述する．

$$p(\phi_n|\phi_{n-1}) = \pi_{\phi_{n-1}\mathrm{mod}16,\phi_n-\phi_{n-1}}, \tag{2}$$

ここで，$\pi_{a,b}$ は，小節内の拍位置 $a\ (0 \le a < 16)$ から開始したコードが．16 分音符 $b\ (0 < b \le T)$ 個分だけ持続する確率を表す．

同様に，メロディのリズム $\boldsymbol{\psi}$ に関するマルコフモデルも以下の遷移確率によって記述する．

$$p(\psi_{n,1}|\psi_{n-1,I_{n-1}}) = \rho_{\psi_{n-1,I_{n-1}}\mathrm{mod}16,\psi_{n,1}-\psi_{n-1,I_{n-1}}},$$
$$p(\psi_{n,i}|\psi_{n,i-1}) = \rho_{\psi_{n,i-1}\mathrm{mod}16,\psi_{n,i}-\psi_{n,i-1}}\ (1 < i), \tag{3}$$

ここで，$\rho_{a,b}$ は，$\pi_{a,b}$ と同様の確率を表す．

コード記号 $\boldsymbol{z}$ の条件のもとでの音高系列 $\boldsymbol{x}$ に関するマルコフモデルは，以下の遷移確率によって記述する．

$$p(p_{n,1}|x_{n-1,I_{n-1}}, z_n) = \tau^{z_n}_{x_{n-1,I_{n-1}},x_{n,1}}, \tag{4}$$
$$p(x_{n,i}|x_{n,i-1}, z_n) = \tau^{z_n}_{x_{n,i-1},x_{n,i}}\ (2 \le i \le I_n), \tag{5}$$

ここで，$\tau^{c}_{a,b}$ は，コード記号 c のもとで音高が a から b に遷移する確率を表す．

統合モデルをベイズ推論可能にするため，各パラメータに対して以下のようなディリクレ（ベータ）共役事前分布をおく．

$$\boldsymbol{\theta}_A \sim \mathrm{Dir}(\boldsymbol{\xi}_A),\ \boldsymbol{\eta}_A \sim \mathrm{Dir}(\boldsymbol{\zeta}_A),\ \lambda_A \sim \mathrm{Beta}(\boldsymbol{\iota}_A), \tag{6}$$
$$\boldsymbol{\pi}_a \sim \mathrm{Dir}(\boldsymbol{\beta}_a),\ \boldsymbol{\rho}_a \sim \mathrm{Dir}(\boldsymbol{\gamma}_a),\ \boldsymbol{\tau}^c_a \sim \mathrm{Dir}(\boldsymbol{\delta}^c_a), \tag{7}$$

ここで，$\boldsymbol{\xi}_A$，$\boldsymbol{\zeta}_A$，$\boldsymbol{\iota}_A$，$\boldsymbol{\beta}_a$，$\boldsymbol{\gamma}_a$，$\boldsymbol{\delta}^c_a$ は，ハイパーパラメータである．

4.3 LSTM に基づく音高遷移モデル

メロディの一部を更新する操作において，長期的で複雑なメロディの遷移を捉えるために，LSTM モデルを用いる．ここで，メロディ全体に関して，$\boldsymbol{p} = \{p_t\}_{t=1}^{T}$ という時間グリッド単位の別の表記法を導入する．p_t は，メロディの冒頭から 16 分音符 $t\ (0 \le t < T)$ 個分の位置から開始する音符があった場合，その音符の MIDI ノートナンバーを表し，そうでない場合には 0 をとるとする．同様

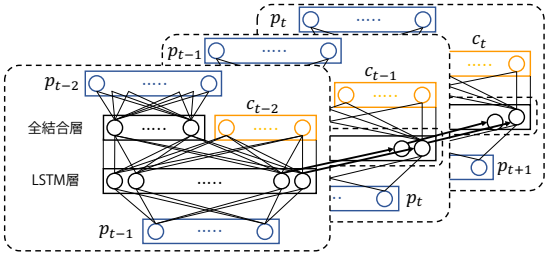

図 3: 本稿で用いる LSTM のネットワーク構成

に，コード進行に関しても，$\boldsymbol{z}$ と $\boldsymbol{\phi}$ の代わりに $\boldsymbol{c} = \{c_t\}_{t=1}^{T}$ と別の形式で表記する．c_t は，メロディの冒頭から 16 分音符 t 個分の位置でのコード記号を表す．与えられたメロディ $p_{1:t} = \{p_i\}_{i=1}^{t}$ 及びコード進行 $c_{1:t} = \{c_i\}_{i=1}^{t}$ をもとに，我々は，次の音符 p_{t+1} の確率 $p(p_{t+1}|p_{1:t}, c_{1:t})$ を図 3 に示す LSTM を用いて学習・生成する．

4.4 ベイズ推定

我々は，モデルパラメータ $\boldsymbol{\Theta} = \{\boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\lambda}, \boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\tau}\}$ を最大事後分布 (MAP) 推定の枠組みから学習する．コード記号列 $\boldsymbol{x}$ から，PCFG のパラメータ $\boldsymbol{\theta}$，$\boldsymbol{\eta}$，$\boldsymbol{\lambda}$ を教師なしで推定するには，ギブスサンプリングの一種である内側フィルタリング・外側サンプリングアルゴリズム [3, 23] を用いる．

マルコフモデルのパラメータ $\boldsymbol{\pi}$，$\boldsymbol{\tau}$，$\boldsymbol{\rho}$ は，教師あり学習によって独立に推定される．$\boldsymbol{\phi}$ が与えられたとき，$\boldsymbol{\pi}$ の事後分布の算出はディリクレ分布とカテゴリカル分布の共役性から容易であり，同様に，$\boldsymbol{z}$，$\boldsymbol{\phi}$，$\boldsymbol{x}$ のデータの組が与えられたとき，$\boldsymbol{\tau}$ の事後分布も求められる．

5. コードとメロディのアレンジ手法

本章では，4 章で述べた階層モデルを用いて，提案システム（図 3）に実装したコード進行とメロディに関する 5 つのアレンジ操作を実現するための手法について述べる．

5.1 コード記号列の更新

メロディ $\boldsymbol{x}$，$\boldsymbol{\psi}$ とコードの開始位置 $\boldsymbol{\phi}$ を固定したとき，コード記号列 $\boldsymbol{z}$ 及びその背後の木構造 $\boldsymbol{t}$ は，条件付き事後分布 $p(\boldsymbol{t}, \boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\Theta})$ を最大化することによって，最適化することができる．まず，終端記号の葉ノード $\boldsymbol{z}$ から頂点ノード S に向かって内側確率を以下のように再帰的に計算する．

$$p_{n,n}^{A} = \lambda_A \max_{z \in \Sigma} \eta_{A \to z}\ p(\boldsymbol{x}_n|z), \tag{8}$$

$$p_{n,n+k}^{A} = (1-\lambda_A) \max_{\substack{B,C \in V \\ 1 \le l \le k}} \theta_{A \to BC} p_{n,n+l-1}^{B} p_{n+l,n+k}^{C}, \tag{9}$$

ここで，$p(\boldsymbol{x}_n|z_n)$ は，コード記号 z_n に条件付けられて音高の部分列 $\boldsymbol{x}_n$ が生成される確率であり，学習したマルコフモデルに基づく音高遷移モデルによって算出される．

$$p(\boldsymbol{x}_n|z_n) = \prod_{i=1}^{I_n} p(x_{n,i}|x_{n,i-1}, z_n), \tag{10}$$

ここで，$x_{n,0} = x_{n-1,I_{n-1}}$ である．最尤の $\boldsymbol{t}$ 及び $\boldsymbol{z}$ は，頂点ノード S から最尤のパスをたどることによって得られる．

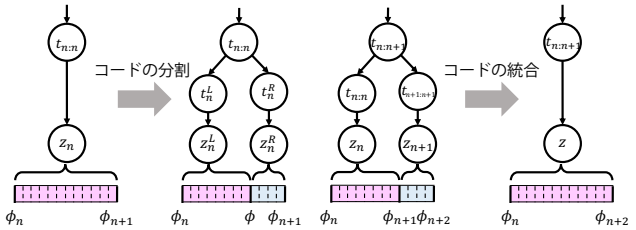

図 4: コード記号の分割操作と統合操作

5.2 コードの開始位置の更新

メロディ $\boldsymbol{x}$，$\boldsymbol{\psi}$ とコードの記号列 $\boldsymbol{z}$ を固定したとき，n 個目のコードの開始位置 ϕ_n を条件付き事後分布

$$p(\phi_n|\boldsymbol{z}, \boldsymbol{\phi}_{\neg n}, \boldsymbol{x}, \boldsymbol{\psi}, \boldsymbol{\Theta}) \propto p(\boldsymbol{x}_{n-1}|z_{n-1}) p(\boldsymbol{x}_n|z_n) p(\phi_n|\phi_{n-1}) p(\phi_{n+1}|\phi_n), \tag{11}$$

を最大化することで最適化することができる．ここで，ϕ_n は，$\psi_{n-1,1} \le \phi \le \psi_{n,I_n}$ の制約下で推定される．

5.3 コードの分割・統合

コード記号列 $\boldsymbol{z}$ とその開始位置 $\boldsymbol{\phi}$ に関して，1 つのコードを隣り合う 2 つのコードの分割したり，隣り合った 2 つのコードを 1 つのコードに統合したりすることもできる（図 4）．分割操作は，コード進行中のいかなるコード z_n にも適用することができるが，統合操作は，部分木 $t_{n:n+1}$ をなす連続したコード $z_{n:n+1}$ にのみ適用することができる．

n 個目のコード z_n 及びそれぞれの上部に与えられた非終端記号 $t_{n:n}$ は，新たなビート位置 $\phi \in (\phi_n, \phi_{n+1})$ で，新たなコード z_n^L，z_n^R と，非終端記号 t_n^L，t_n^R に分割される．推定するべき 5 変数は，条件付き事後分布 $p(t_n^L, t_n^R, z_n^L, z_n^R, \phi | t_{\neg n:n}, \boldsymbol{z}_{\neg n}, \boldsymbol{\phi}, \boldsymbol{x}, \boldsymbol{\psi}, \boldsymbol{\Theta})$ を最大化するものが選ばれる．この操作を行うために，$\boldsymbol{x}_n$ をもとに最尤の部分木 $t_{n:n}$ を求めるビタビアルゴリズムを用いる．まず，内側確率を部分木 $t_{n:n}$ の終端記号である z_n^L，z_n^R から頂点ノード $t_{n:n}$ に向けて以下のように再帰的に計算される．

$$\alpha_{\phi}^{A} = \lambda_A \max_{z \in \Sigma} \eta_{A \to z}\ p(\boldsymbol{x}_n^L|z, \phi), \tag{12}$$

$$\beta_{\phi}^{A} = \lambda_A \max_{z \in \Sigma} \eta_{A \to z}\ p(\boldsymbol{x}_n^R|z, \phi), \tag{13}$$

$$p_{\phi}^{t_{n:n}} = \max_{B,C \in V} \theta_{t_{n:n} \to BC} \alpha_{\phi}^{B} \beta_{\phi}^{C} p(\phi|\phi_n) p(\phi_{n+1}|\phi), \tag{14}$$

ここで，$\boldsymbol{x}_n^L$ 及び $\boldsymbol{x}_n^R$ は，境界 ϕ のもとで $\boldsymbol{x}_n$ を分割することによって得られる音高部分列を表す．最尤の z_n^L，z_n^R，t_n^L，t_n^R，ϕ は，$t_{n:n}$ から最尤のパスを再帰的にたどることによって得られる．

隣り合う 2 つのコード z_n，z_{n+1} 及びその上部にある非終端記号 $t_{n:n}$，$t_{n+1:n+1}$ は，単一のコード z 及び新たな非終端記号 $t_{n:n+1}$ に統合される．z は，条件付き事後分布 $p(z|t_{\neg n:n+1}, \boldsymbol{z}_{\neg n:n+1}, \boldsymbol{\phi}_{\neg n+1}, \boldsymbol{x}, \boldsymbol{\psi}, \boldsymbol{\Theta})$ を最大化するものが選ばれる．最尤の z は，次式のように選ばれる．

$$z = \arg\max_{z' \in \Sigma} \eta_{t_{n:n+1} \to z'}\ p(\boldsymbol{x}_n|z') p(\boldsymbol{x}_{n+1}|z'). \tag{15}$$

5.4 メロディの更新

n 番目のコード記号 z_n と，その直前の音符 $x_{n-1,I_{n-1}}$，そしてその直後の音符 $x_{n+1,1}$ が与えられたとき，コード z_n の支配区間 $[\phi_n, \phi_{n+1})$ 中の音符を，条件付き事後分布 $p(\boldsymbol{x}_n|z_n, x_{n-1,I_{n-1}}, x_{n+1,1}, \boldsymbol{\Theta})$ にしたがって，サンプリングを行うことで更新する．まず，拍位置 $\phi_n + t$ に音高 y_t の音符があり，その直前の音符の長さが d_t であるときの確率 $\alpha_{y_t,d_t} = p(y_t, d_t|z_n)$ をビート位置 $t \in \{\phi_n, ..., \phi_{n+1}, \psi_{n+1,1}\}$ において以下のように再帰的に計算する．

$$\alpha_{y_t,d_t} = \rho_{t-d_t,t} \sum_{y_{t-d_t},\, d_{t-d_t}} \alpha_{y_{t-d_t},d_{t-d_t}} \tau^{z_n}_{y_{t-d_t},y_t}$$

ただし，各 t で d_t は，$d_t \in \{1, ..., t, t - \psi_{n-1,I_{n-1}}\}$ の状態空間を持つ．この確率を用いて，$\psi_{n+1,1}$ から $\psi_{n-1,I_{n-1}}$ に向かって，音高 y_t と直前の音符の長さ d_t を 1 組ずつサンプリングしていくことで，新たな音高列 $\boldsymbol{x}_n$，$\boldsymbol{\psi}_n$ を得る．

さらに，以上のマルコフモデルに基づくメロディの部分生成手法の改良手法として，LSTM を用いた手法についても提案する．メロディ全体 $\boldsymbol{p}$ のうち，部分列 $p_{i:j}$ を更新すると仮定すると，部分列 $p_{i:j}$ は，与えられたコード進行 $\boldsymbol{c}$ 及び $p_{1:i-1}$ と $p_{j+1:T}$ に対して，条件付き事後分布 $p(p_{i:j}|\boldsymbol{c}, p_{1:i-1}, p_{j+1:T}) \propto p(\boldsymbol{p}|\boldsymbol{c})$ にしたがってサンプリングできる．具体的には，まず，更新箇所以前のメロディ $p_{1:i-1}$ 及びコード $c_{1:i-1}$ をネットワークに入力し，ネットワークの各層を更新する．次に，更新するべきメロディ $p_{i:j}$ を，学習後の LSTM の出力から得られる遷移確率 $p(p_{t+1}|p_{1:t}, c_{1:t})$ によって逐次的にサンプリングする．また，同様の遷移確率を更新した箇所を含むメロディ部分列 $p_{i:T}$ に対して求め，それらを掛け合わせることで，メロディ全体の事後確率 $p(\boldsymbol{p}|\boldsymbol{c})$（に比例する値）が算出できる．以上の方法によって得られたサンプルを十分な数生成し，求めた確率 $p(\boldsymbol{p}|\boldsymbol{c})$ を最大化するサンプルを新たな $p_{i:j}$ として決定する．

6. 評価実験

本章では，提案システムに対する定量的評価及び定性的評価の結果について述べる．

6.1 実験条件

コード記号生成モデル (PCFG) の学習には，The SALAMI Annotation Data [24] 内のポピュラー音楽 468 曲から抽出した，A メロなどの楽節に対応する 8 小節分のコード進行 705 個を用いた．コード記号は，ルート音 {C, C#, ..., B} と {major, minor} の組み合わせの 24 種とした．マルコフモデルの学習には，Rock Corpus [25] 内のポピュラー音楽 194 曲から抽出した，コード進行とメロディのペア 9902 組を用いた．なお，すべての楽曲の調はすべて C に移調した．また，PCFG の非終端記号数は 15 とした．各パラメータのハイパーパラメータの値は，$\boldsymbol{\lambda}$ の場合は 1.0，それ以外の場合は 0.1 とした．LSTM の学習には，Rock Corpus と Nottingham Database から抽出した，コード進行とメロディのペア 9265 組を用いた．隠れ層の数は 50 で，損失関数には softmax-cross-entropy を用い，パラメータ更新には，Adam を用いた．5.4 章で述べた LSTM に基づく手法でメロディを更新する場合に生成するサンプル数は 50 とした．

6.2 メロディ更新手法に関する客観的評価

5.4 章で提案した 2 種類のメロディ更新手法を定量的に評価するために，部分生成されたメロディの音符の密度と，それ以外の箇所の音符の密度との比較を Rock Corpus と Nottingham Database のデータにおける 10 分割の交差検証のもと行った．具体的には，それぞれのコード z_n の支配区間において，メロディ部分列 $\boldsymbol{p}_n$ をマルコフモデルと LSTM に基づく 2 種類のメロディ更新手法を用いて更新する．そして，部分生成された箇所 $[\phi_n, \phi_{n+1})$ における 1 小節ごとの音符数と，その他の箇所における 1 小節ごとの音符数の間の平均二乗誤差 (MSE) を以下のように求める．

$$\mathrm{MSE} = \frac{1}{N-1} \sum_{n=1}^{N-1} \left\{ \frac{16 I_n^*}{\phi_{n+1} - \phi_n} - \frac{\sum_{m \neq n} 16 I_m}{\sum_{m \neq n} (\phi_{m+1} - \phi_m)} \right\}^2,$$

ここで，I_n^* は，メロディの更新操作によって更新された音符数を表し，I_n は，更新前のメロディの $\boldsymbol{x}_n$ に含まれる音符数である．平均二乗誤差は，データのメロディに関して，またすべてのコード区間において求めた．平均二乗誤差の値は，LSTM に基づく手法の場合には 5.52 であり，マルコフモデルに基づく手法の場合には 6.42 であった．この結果から，LSTM に基づく手法の方がメロディの長期的な特徴を考慮できるため，マルコフモデルに基づく手法よりメロディ全体の音符の密度を考慮してメロディを更新する能力が少し高いことが明らかになった．

6.3 提案システムに対する主観評価

我々は，提案システムにおける対話的なコードとメロディの編集動作の性能について評価するためにユーザテストを行った．RWC データベースから抽出した 8 小節からなる 5 種類のメロディを用いて，11 人の被験者に提案システムを試用してもらった．11 人のうち 4 人は音楽的知識（5 年以上の楽器演奏経験）のある被験者であった．被験者には，5 種類のメロディを用いて対話的に楽曲を編集してもらい，その後，以下の 14 個の指標において，5 段階のリッカート尺度（(1) まったくそう思わない，(2) あまりそう思わない，(3) どちらとも思わない，(4) ややそう思う，(5) とてもそう思う）による評価を依頼した．

- 得られたコード進行はメロディに対してふさわしかった (I).
- 分割操作または統合操作を行うことによって得られたコード進行は自然だった (Ⅱ, Ⅲ).
- 更新後のメロディはコードに対して自然だった (Ⅳ).

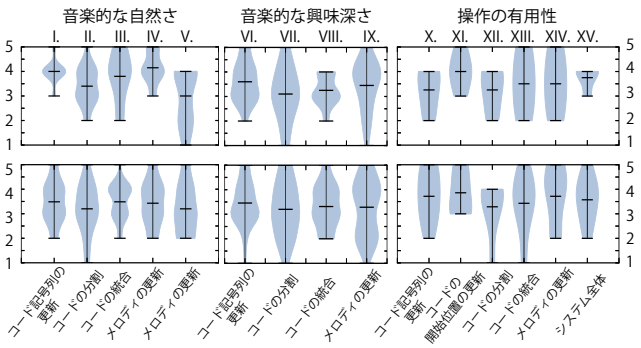

図 5: 被験者実験の結果．上段が音楽的知識のある被験者の評価結果．下段がそれ以外の被験者の評価結果．

- 更新後のメロディは全体として自然だった (V).
- ４操作（コード進行の更新，コードの分割，コードの統合，メロディの更新）によって得られた楽曲に興味深いものがあった (VI, VII, VIII, IX).
- ４操作は有用であった (X, XI, XII, XIII).
- 提案システムは，ユーザが作曲や編集する上で有用性があった (XIV).

被験者実験の結果を図 5 に示す．生成結果の音楽的な自然さと，興味深さという観点では，コード進行の更新操作とメロディの更新操作への評価は，指標 (I)，(IV)，(V) においてそれぞれ平均 3.67，3.69，3.51 とやや高い評価を得た．指標 (V)，(IX) に注目すると，音楽的知識のある被験者の方が，それ以外の被験者よりもメロディの更新操作による生成結果が自然だと感じる割合が少なかったが，興味深いと感じる割合が多いという知見が得られた．また，各操作及びシステム全体の有用性に関しては，各操作が平均得点が最小で 3.27，最大で 3.91 というやや高い評価結果を得た．

さらに，提案システムの使用性に関して以下のような意見を得た．

- 作曲経験がなくとも，複数の操作を組み合わせることで楽曲が編集できるのが興味深かった．
- 音楽的素養がなくとも，作曲をしているような気分になれて楽しかった．
- コード進行の編集をしやすくするため，単に単一のコード記号を更新するような操作が欲しかった．

また，各操作における問題について以下のような意見を得た．

- 得られたコード進行はたいてい適切であったが，シンプルなコード記号（C メジャー，F メジャー）を生成する傾向があった．
- 繰り返し箇所のあるメロディの一部を更新した場合に，不自然に聞こえる場合があった．

前者の問題は，コード記号の更新操作をビタビアルゴリズムによって実現しているためであると考えられる．後者の問題は，LSTM がメロディ全体の繰り返し構造を考慮できていないためであると考えられる．以上から，各操作にいくつか改善点を残しながらも，提案システムの対話的なコードとメロディの自動編集操作に，ユーザの楽曲制作をサポートする上での有用性があるとわかった．

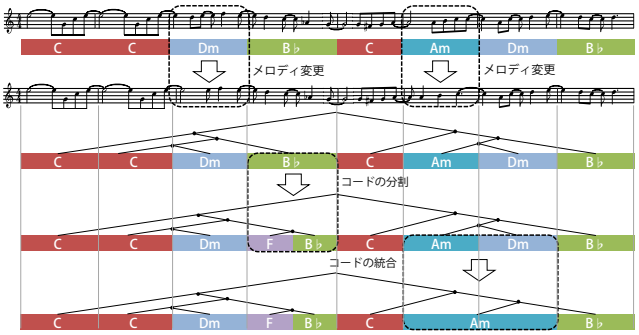

図 6: 提案手法を用いたコード進行とメロディの生成例

6.4 システム動作例・生成例

提案システムを用いたコード進行とメロディの対話的生成の例を図 6 に示す．図中の 1 段目は，はじめに与えたメロディとそれに対して生成されたコード進行を示す．ただし，コードの開始位置は小節線の位置に置いた．2 段目は，3，6 個目のコードのもとでメロディを更新した状態を示す．3 段目は，4 個目のコード (1 小節) を分割した後の結果を示す．4 段目は，7，8 個目のコードを統合した後の結果を示す．以上から，提案システムによって，コード進行の背後にある木構造を考慮しながらコード進行を生成・操作したり，メロディ全体の一貫性を保ちながらメロディの一部を更新したりすることができることが明らかになった．

7. おわりに

本稿では，コード進行とメロディに関する統一的な確率モデルを用いて，コード進行とメロディを互いに対話的に生成することで楽曲を洗練できるインタラクティブ作曲システムを提案した．また，評価実験によって，提案システムがユーザの楽曲制作を支援するシステムとして高い有用性があることが示された．

しかし，現在のメロディの更新手法では，変更したいメロディの前後のメロディとの連結性は考慮しているが，変更後のメロディの大局的な構造は考慮していない．今後は，変更後のメロディ全体の音楽的妥当性を保証するための手法変更を行いたい．また将来的には，提案システムを用いて大規模なユーザテストを行い，各ユーザの操作履歴のデータをもとに，音楽的な好みを推論し，学習モデルを強化学習によって改善する手法を試みたい．また，同様のデータを用いて，人間が楽曲を洗練させながら創作していく過程を明らかにすることも可能であると考えられる．

謝辞　本研究の一部は、JST ACCEL No. JPMJAC1602, JSPS 科研費 No. 26700020，No. 16H01744，科研費 特別研究員奨励費 No. 16J05486 の支援を受けた．

参考文献

[1] Chuan, C. H. and Chew, E.: A hybrid system for automatic generation of style-specific accompaniment, *IJWCC*, pp. 57–64 (2007).
[2] Simon, I., Morris, D. and Basu, S.: MySong: automatic accompaniment generation for vocal melodies, *SIGCHI Conference on Human Factors in Computing Systems*, ACM, pp. 725–734 (2008).
[3] Tsushima, H., Nakamura, E., Itoyama, K. and Yoshii, K.: Function- and rhythm-aware melody harmonization based on tree-structured parsing and split-merge sampling of chord sequences, *ISMIR*, pp. 502–508 (2017).
[4] Prisco, R. D., Eletto, A., Torre, A. and Zaccagnino, R.: A neural network for bass functional harmonization, *European Conference on the Applications of Evolutionary Computation*, Springer, pp. 351–360 (2010).
[5] Fukayama, S. et al.: Orpheus: Automatic composition system considering prosody of Japanese lyrics, *ICMC*, Springer, pp. 309–310 (2009).
[6] Roig, C., Tardón, L. J., Barbancho, T. and Barbancho, A. M.: Automatic melody composition based on a probabilistic model of music style and harmonic rules, *Knowledge-Based Systems*, Vol. 71, pp. 419–434 (2014).
[7] Steedman, M. J.: A generative grammar for jazz chord sequence, *Music Perception*, Vol. 2, No. 1, pp. 52–77 (1984).
[8] Rohrmeier, M.: Mathematical and computational approaches to music theory, analysis, composition and performance, *Journal of Mathematics and Music*, Vol. 5, No. 1, pp. 35–53 (2011).
[9] Raczyński, S. A., Fukayama, S. and Vincent, E.: Melody harmonization with interpolated probabilistic models, *Journal of New Music Research*, Vol. 42, No. 3, pp. 223–235 (2013).
[10] Ebcioğlu, K.: An expert system for harmonizing four-part chorales, *Computer Music Journal*, Vol. 12, No. 3, pp. 43–51 (1988).
[11] Papadopoulos, G. and Wiggins, G.: AI methods for algorithmic composition: A survey, a critical view and future prospects, *AISB Symposium on Musical Creativity*, pp. 110–117 (1999).
[12] Towsey, M., Brown, A., Wright, S. and Diederich, J.: Towards melodic extension using genetic algorithms, *Educational Technology & Society*, Vol. 4, No. 2, pp. 54–65 (2001).
[13] Prisco, R. D. and Zaccagnino, R.: An evolutionary music composer algorithm for bass harmonization, *Applications of Evolutionary Computing*, Springer, pp. 567–572 (2009).
[14] Allan, M. and Williams, C.: Harmonising chorales by probabilistic inference, *NIPS*, pp. 25–32 (2005).
[15] Groves, R.: Automatic harmonization using a hidden semi-Markov model, *AIIDE*, pp. 48–54 (2013).
[16] Paiement, J. F., Eck, D. and Bengio, S.: Probabilistic melodic harmonization, *CSCSI*, pp. 218–229 (2006).
[17] Hadjeres, G. and Pachet, F.: DeepBach: A steerable model for Bach chorales generation, *ICML*, pp. 1362–1371 (2017).
[18] Waite, E.: Generating long-term structure in songs and stories, https://magenta.tensorflow.org/2016/07/15/lookback-rnn-attention-rnn.
[19] Yang, L. C., Chou, S. Y. and Yang, Y. H.: MidiNet: A convolutional generative adversarial network for symbolic-domain music generation, *ISMIR*, pp. 324–331 (2017).
[20] Mogren, O.: C-RNN-GAN: Continuous recurrent neural networks with adversarial training, *Constructive Machine Learning Workshop (NIPS 2016)* (2016).
[21] Boulanger-Lewandowski, N., Bengio, Y. and Vincent, P.: Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription, *ICML* (2012).
[22] Eck, D. and Schmidhuber, J.: A first look at music composition using LSTM recurrent neural networks, *IDSIA*, Vol. 103, No. 07-02 (2002).
[23] Johnson, M., Griffiths, T. L. and Goldwater, S.: Bayesian inference for PCFGs via Markov chain Monte Carlo, *NAACL-HLT*, pp. 139–146 (2007).
[24] Smith, J. B. L., Burgoyne, J. A., Fujinaga, I., Roure, D. D. and Downie, J. S.: Design and creation of a large-scale database of structural annotations, *ISMIR*, pp. 555–560 (2011).
[25] Clercq, T. D. and Temperley, D.: A corpus analysis of rock harmony, *Popular Music*, Vol. 30, No. 01, pp. 47–70 (2011).