

[ポスター講演] 和音系列に対する PCFG のベイズ学習と Split-Merge サンプリングを用いたメロディへの和声付け

津島啓晃[†] 中村 栄太[†] 糸山 克寿[†] 吉井 和佳^{†, ††}

[†] 京都大学 大学院情報学研究科

^{††} 理研 AIP

E-mail: †{tsushima, enakamura, itoyama, yoshii}@sap.ist.i.kyoto-u.ac.jp

あらまし 本稿では、音楽コーパスから和音系列とメロディに関する生成規則を統計的に学習し、それに基づいてメロディへの和声付けを行う手法を示す。従来の和声付け手法には、一拍ごとのコードの遷移を表現した隠れマルコフモデル (HMM) に基づく手法がある。しかしこの手法では、音楽理論において重要とされているコードのリズム、コードの機能 (tonic, dominant, subdominant), コードの階層構造を明示的に表現できない。この問題を解決するため、確率的文脈自由文法による和音系列生成モデル、拍節マルコフモデルによるコードのリズム生成モデル、コードの条件付きマルコフモデルによる音高系列生成モデルからなる階層的生成モデルを提案する。さらに、提案モデルを用いてメロディに対する和音系列の推定を行うため、潜在変数であるコード記号とそのオンセット位置のそれぞれを split-merge サンプリングという新しいサンプリング手法を含むメトロポリス・ヘイスティングス法に基づいて更新する手法を提案する。評価実験より HMM に基づく手法に対して提案手法の和声付けタスクにおける精度が向上したことを示せた。キーワード 和音系列, コード進行, 和声付け

1. はじめに

コード進行は、特定の音楽スタイルや楽曲の持つ印象を特徴付けることから、コード進行の決定は、作曲・編曲において重要な役割を担っている。本研究では、入力メロディ (音符系列) に対してコード記号系列を自動生成することを目的とした自動和声付け問題に取り組む。本稿では、ポピュラー音楽のスタイルでの自動和声付けに焦点を当てる。我々は、和音系列に関する音楽理論を人手で記述するかわりに、音楽コーパスから和音系列とメロディの生成規則を表すモデルパラメータを統計的に学習する。そのために、和音系列の出現しやすさを表現する確率モデルと、音符系列の出現しやすさを表現する確率モデルをそれぞれ提案する。

和音系列は通常、マルコフモデルによって表現されるため [1, 2], 標準的な自動和声付け手法では、コードを隠れ状態、音高を出力記号とした隠れマルコフモデル (HMM) が用いられる。しかし、この手法ではコードの構文的役割や階層的構造が考慮されていない。構文的役割は、伝統的な音楽理論 [3, 4] では機能と声という言語理論における品詞のような概念として言及されている。さらに、この手法には、コードの継続長が隠れ状態 (コード記号) の自己遷移によって表現されるため、コードのリズムが明示的に表現されない、また、音高がコードから生成されると仮定しているため、音高遷移が考慮されないという問題がある。

以上の問題を解決するため、我々は、(1) 確率的文脈自由文法 (PCFG) に基づくコード記号系列生成モデル、(2) 拍節マルコフモデルに基づくコードのリズム生成モデル、(3) コードの

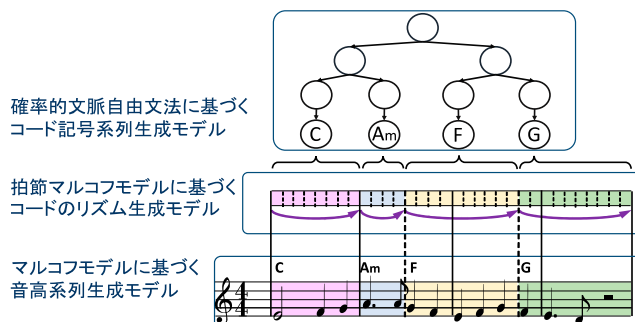


図 1: コード記号, コードのリズム, メロディに関する階層的生成モデル

条件付きマルコフモデルに基づく音符系列生成モデルからなる階層的生成モデルを提案する (図 1)。コード記号系列生成モデルに PCFG を用いるのは、Steedman [5] がコード記号系列の階層的構造を文脈自由文法 (CFG) を用いて表現したことから着想を得ている。このモデルの長所は、PCFG のルール確率が音楽コーパス内の和音系列から教師なしで推定され、その際、PCFG の非終端記号によって上述のコードの構文的役割が捉えられることが期待できる点である。また、拍節マルコフモデルは、連続するコードのオンセットビート位置の遷移確率を明示的に表現している。

さらに我々は、提案した階層的生成モデルと、split-merge サンプリングという新しいサンプリング手法を含むメトロポリス・ヘイスティングス (MH) 法を用いた、和声付け問題への統計的手法を提案する。本手法では、与えられたメロディに対し

て可変長の和音系列とそのリズムを推定するために、メロディに対して最尤の和音系列とその背後にある木構造、そしてそのオンセットビート位置をその事後分布に基づいて確率的に探索する。探索には、次の4種の提案分布を用いたサンプラーを用いる。(1)コードとその背後にある木構造は、ビタビアルゴリズムを拡張したアルゴリズムを用いて更新する。コードの数は、(2)一つのコードを二つに分割する、もしくは(3)隣り合う二つのコードをマージすることによって局所的に更新する。(4)コードのリズムは、一つのコードのオンセットビート位置を前後に移動することによって局所的に更新する。これらの更新動作は、人間が最適な和音系列を試行錯誤的に探索する操作としても解釈できる。

2. 関連研究

本章では自動和声付けと音楽言語モデルに関する関連研究をそれぞれ紹介する。

2.1 自動和声付け

自動和声付けに関する研究は、本研究を含むコード記号系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、ChuanとChew [6]は、入力メロディに対して、コードのルート音をサポートベクターマシンによって選択し、ルート音から三和音とコード進行をルールベースで生成するという複合的な手法を提案した。Simonら [1]は、コードの遷移をマルコフモデルによって表現したHMMに基づく手法を提案した。この手法は、商用の自動和声付けシステム*MySong*に実装された。Raczyńskiら [2]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よる手法を提案した。本研究のように、PCFGを用いた自動和声付け手法は今までにない試みといえる。

後者の方向性では、Ebcioğlu [7]は、パツ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 [8–10]も多く行われている。AllanとWilliamsら [11]は、コードを隠れ状態、音符を出力記号として表現した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)を用いた研究もなされている [12]。Paiementら [13]は、コード記号表記を分割し、それぞれ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。

2.2 音楽言語モデル

音楽の構造解析を目的とした音符の言語モデルに関しては様々な研究がなされている [14–17]。生成音楽理論 (GTTM) [14]では、音符系列は、複数の音符列からどの音符がより重要であるかにしたがって、音符の簡約化ができるという観点から、その背後に階層構造を持っていると仮定し、その規則を手手で記述している。GTTMの計算的実装や楽曲解析への利用についても研究されている [15, 16]。また、GTTMをPCFGを用いて確率モデルとして拡張し、音符系列のみから音符の生成規則を教師なし学習することも試みられている [17]。

和音系列に対しても、様々な言語モデルが提案されている。

その関連タスクは、音楽音響信号の和音認識 [18–20]、音楽解析 [5, 21]、編曲 [22, 23]と多岐にわたる。従来の和音系列の言語モデルには、 n -gramモデルが挙げられる [18, 23]。一般的に、 n -gramモデルには次数を増やせば学習データがスパースになるゼロ確率問題が存在するが、言語モデルの予測性能を向上するために適切なスムージング手法も検討されている [24]。Yoshiiら [19]は、各コードがそれぞれ任意長のコンテキストを考慮することを可能とする語彙フリー無限グラムモデルを提案している。Paiementら [20]は、コードの階層的な構造を表現するために複数の隠れ層による状態遷移モデルを提案した。また、コードの階層的構造を表現する文法的規則を明示的に記述しようとする研究もなされている [5, 21, 22]。Steedman [5]とRohrmeier [21]は、和音系列に関する生成規則を記述することを提案した。のちに、確率モデルへの拡張が提案され、そのルール確率の教師なし学習に関する研究もなされた [22]。これらの研究では、具体的な非終端記号や生成規則は、音楽理論や感性的な直観から人手で与えられた。

3. 提案モデル

本章では、コードとメロディに関する階層的生成モデルの定式化と学習方法について述べる。コード記号系列の生成モデルであるPCFGは、音楽コーパス内のコード記号系列データから教師なし学習し、コードのリズムに関する拍節マルコフモデルは、コードのリズムデータから教師あり学習し、音高系列に関するマルコフモデルは、和音系列とメロディのペアの学習データから教師あり学習する。

3.1 モデル定式化

PCFGによって確率的に生成されたコード記号系列を $z = \{z_n\}_{n=1}^N$ 、拍節マルコフモデルによって生成された各コードのオンセットビート時刻 (16分音符単位) を $\phi = \{\phi_n\}_{n=1}^N$ と記す。ここで、 N はコードの数である。生成された各コード z_n の支配区間のもとで生成された音高系列の部分列を $\mathbf{x}_n = \{x_{n,i}\}_{i=1}^{I_n}$ と記す。ここで、 I_n は、その区間に存在する音符数である。部分列 \mathbf{x}_n を繋ぎ合わせることで得られる音高系列全体を $\mathbf{x} = \{\mathbf{x}_n\}_{n=1}^N$ と表す。 $I = \sum_{n=1}^N I_n$ は、メロディ中の音符の総数である。 $\psi_{n,i}$ は、音符 $x_{n,i}$ のオンセットビート時刻を表し、その集合を $\psi = \{\{\psi_{n,i}\}_{i=1}^{I_n}\}_{n=1}^N$ とする。 ϕ_n と $\psi_{n,i}$ は、0から $16L - 1$ の整数値を取りうる (L はメロディの小節数)。本章では、簡単のため、単一系列の場合について定式化を行ったが、実際に提案モデルを学習するときには、複数の和音系列やメロディを取り扱い、その際の拡張は容易である。

本稿で用いる確率的文脈自由文法 G を以下のように定義する。

$$G = (V, \Sigma, R, S) \quad (1)$$

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。 $\theta_{A \rightarrow BC}$ 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、 $\eta_{A \rightarrow \alpha}$ 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 $A (\in V)$ 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。

$$\sum_{B,C \in V} \theta_{A \rightarrow BC} = 1 \quad \sum_{\alpha \in \Sigma} \eta_{A \rightarrow \alpha} = 1 \quad (2)$$

また、 $\theta_A = \{\theta_{A \rightarrow BC}\}_{B,C \in V}$, $\eta_A = \{\eta_{A \rightarrow \alpha}\}_{\alpha \in \Sigma}$, $\theta = \{\theta_A\}_{A \in V}$, $\eta = \{\eta_A\}_{A \in V}$ を定義する。同様の表記法を本稿を通して用いる。

コードのリズムに関する拍節マルコフモデルは、コードの16分音符単位でのオンセットビート位置 (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。

$$p(\phi_n | \phi_{n-1}) = \pi_{\bar{\phi}_{n-1}, \bar{\phi}_n} \quad (3)$$

ここで、 $\bar{\phi}_n (= \phi_n \bmod 16)$ 及び π_{ab} ($0 \leq a, b < 16$) は、小節内の拍位置が、 k から l に遷移する確率を表す。ただし、 $\bar{\phi}_n \leq \bar{\phi}_{n-1}$ であるときには、コード z_{n-1} が小節線をまたいでいると解釈する。

音高系列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。

$$p(x_{n,m} | x_{n,m-1}, z_n) = \tau_{x_{n,m-1}, x_{n,m}}^{z_n} \quad (4)$$

ここで、 $\tau_{x_{n,m-1}, x_{n,m}}^{z_n}$ は、和音 z_n において、音高 $x_{n,m-1}$ から $x_{n,m}$ に遷移する確率を表す。さらに、 z_{n-1} と z_n の境界での音高遷移確率 $p(x_{n,1} | x_{n-1, l_{n-1}}, z_n)$, z_1 の支配区間における初期確率 $p(x_{1,1} | z_1)$ も導入する。

PCFGのパラメータに対して以下のようなディリクレ共役事前分布をおく。

$$\theta_A \sim \text{Dirichlet}(\xi_A) \quad (5)$$

$$\eta_A \sim \text{Dirichlet}(\zeta_A) \quad (6)$$

$$\lambda_A \sim \text{Beta}(\iota_A) \quad (7)$$

ここで、 ξ_A , ζ_A , ι_A は、ハイパーパラメータである。同様に、マルコフモデルのパラメータに対して、以下のようなディリクレ共役事前分布をおく。

$$\pi_\alpha \sim \text{Dirichlet}(\beta) \quad (8)$$

$$\tau_x^z \sim \text{Dirichlet}(\gamma) \quad (9)$$

ここで、 β と γ は、ハイパーパラメータである。

本来、コードとメロディに関する生成モデルを完結させるには、音符のオンセット位置 ψ に関する生成モデルを決定する必要があるが、本稿における和声付け問題ではメロディは入力として与えられるため、 ψ に関する生成モデルは本稿では省略する。

3.2 ベイズ推定

我々は、モデルパラメータ $\Theta = \{\theta, \eta, \lambda, \pi, \tau\}$ を最大事後分布 (MAP) 推定の枠組みから学習する。PCFGのパラメータ θ , η , λ を推定するには、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 [25] を用いる。提案モデルでは、和音系列 z は潜在的な木構造 t から生成されると仮定する。 t は、非終端記号 $\{t_{n:m}\}_{1 \leq n \leq m \leq N}$ の集合として表現される。ここで、 $t_{n:m}$ は、和音部分列 $z_{n:m} = \{z_n, z_{n+1}, \dots, z_m\}$ を出力する部分木の根ノードとする。潜在木 t とパラメータ θ , η , λ は、それぞれ条件付き事後分布 $p(t | \theta, \eta, \lambda, z)$, $p(\theta, \eta, \lambda | t, z)$ によって交互にサンブルされる。このアルゴリズムによって、真の事後分布 $p(\theta, \eta, \lambda, t | z)$ に基づいて、 t , θ , η , λ のサンプルが得られることが証明されている。

内側フィルタリングステップでは、和音部分列 $z_{n:m}$ を生成するような部分木の根ノードが A である以下の条件付き確率 (内側確率) に着目する。

$$p_{n,m}^A = p(z_{n:m} | t_{n:m} = A) \quad (10)$$

この確率は、葉ノードから開始記号 (S) に向かって以下のように反復計算することができる。

$$p_{n,n}^A = \lambda_A \eta_{A \rightarrow z_n} \quad (11)$$

$$p_{n,n+k}^A = \sum_{B,C \in V} \left[(1 - \lambda_A) \theta_{A \rightarrow BC} \sum_{1 \leq l \leq k} p_{n,n+l-1}^B p_{n+l,n+k}^C \right]$$

外側サンプリングステップでは、開始記号 S から葉ノードに向かって、 $p(t | \theta, \eta, \lambda, z)$ を内側確率を用いて計算することで、再帰的に潜在木 t をサンブルする。ノード $t_{n:n+k} = A$ が既にサンブルされているとき、 $t_{n:n+k}$ が分岐する非終端記号 B , C は以下のようにサンブルされる。

$$\begin{aligned} p(l, B, C) &= p(t_{n:n+l-1} = B, t_{n+l:n+k} = C | t_{n:n+k} = A, z_{n:n+k}) \\ &= (1 - \lambda_A) \theta_{A \rightarrow BC} p_{n,n+l-1}^B p_{n+l,n+k}^C / p_{n,n+k}^A \end{aligned} \quad (12)$$

ここで、 l ($1 \leq l \leq k$) はノードが分岐する位置である。

次に、パラメータ θ , η , λ を確率 $p(\theta, \eta, \lambda | t, z) = p(\theta | t, z) p(\eta | t, z) p(\lambda | t, z)$ にしたがって次のようにサンブルする。

$$\theta_A \sim \text{Dirichlet}(\xi_A + \mathbf{u}_A) \quad (13)$$

$$\eta_A \sim \text{Dirichlet}(\zeta_A + \mathbf{v}_A) \quad (14)$$

$$\lambda_A \sim \text{Beta}(\iota_A + \mathbf{w}_A) \quad (15)$$

ここで、 $\mathbf{u}_{A \rightarrow BC}$ ($\mathbf{v}_{A \rightarrow \alpha}$) は、サンブルされた木 t について、分岐ルール $\theta_{A \rightarrow BC}$ (出力ルール $\eta_{A \rightarrow \alpha}$) が使われた回数を表し、 $\mathbf{w}_{A,0}$ ($\mathbf{w}_{A,1}$) は、非終端記号 A が分岐 (出力) した回数を表す。

マルコフモデルのパラメータ π , τ は、教師あり学習によって推定される。 ϕ が与えられたとき、 π 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、 z , ϕ , \mathbf{x} のデータの組が与えられたとき、 τ の事後分布も求められる。

4. 提案手法

本章では、3.章で述べた提案モデルを用いて、与えられたメ

ロディに対して和音系列を生成する手法について述べる。

4.1 問題設定

我々の目的は、入力メロディ (音高 \mathbf{x} 及びオンセットビート位置 ψ) に対して、学習パラメータ Θ を用いて、和音系列 \mathbf{z} とその位置 ϕ を推定することである。ただしコードの数 N は固定されず、これに関しても推定する。また、和音系列の背後にある木構造 t に関しても推定する。

4.2 メトロポリス・ヘイスティングス法

我々は、split-merge サンプリングという新たな手法を含む MH 法を用いて、事後分布 $p(t, \mathbf{z}, \phi | \mathbf{x}, \psi, \Theta)$ から潜在変数 t , \mathbf{z} , ϕ を推定する。この手法には、以下の 4 種の提案分布を用いる。

- **コード記号の更新:** コード記号系列 \mathbf{z} とその潜在木 t をコードの位置と数を固定したまま、PCFG におけるビタビアルゴリズムを用いることで更新する。
- **Split 操作:** 一つのコードをランダムに選び、二つのコードに分割する。
- **Merge 操作:** 隣接するコードの組をランダムに選びマージする。
- **コードのリズムの更新:** 一つのコード z_n をランダムに選び、そのオンセットビート位置 ϕ_n を前後に動かす。

本来、コード記号の更新には、内側フィルタリング・外側サンプリングを用いるべきだが、真の事後分布を効率的に探索するためにビタビアルゴリズムを用いた。

本研究における MH サンプラーは、4 種の提案分布から一つをランダムに選択し、現在のサンプル $s = (t, \mathbf{z}, \phi)$ から、新たなサンプル $s^* = (t, \mathbf{z}, \phi)^*$ を提案し、以下のように採択率を計算しサンプルの受理・棄却を決定する。

$$g(s^*, s) = \min \left\{ 1, \frac{p(s^*)p(s|s^*)}{p(s)p(s^*|s)} \right\} \quad (16)$$

ここで、 $p(s)$ は、提案モデルにおける完全データ尤度を表し、 $p(s^*|s)$ は、上述の提案分布を表す。提案されたサンプルが棄却された場合には、潜在変数 t , \mathbf{z} , ϕ は更新されない。提案手法では、与えられた \mathbf{x} , ϕ に対して、MH サンプラーを十分な回数試行し、完全データ尤度を最大化する潜在変数を得ることで、最尤な \mathbf{z} , ϕ を推定する。

4.3 コード記号の更新

コード記号 \mathbf{z} とその背後の木構造 t を事後分布 $p(t, \mathbf{z} | \phi, \mathbf{x}, \psi, \Theta)$ にしたがって更新する手法を説明する。

4.3.1 ビタビアルゴリズム

音高 \mathbf{x} 及びオンセットビート位置 ϕ が与えられたとき、コード記号 \mathbf{z} と潜在木 t のサンプルをビタビアルゴリズムを用いて効率的に求める。ただし、コード記号 \mathbf{z} も潜在状態となっているため、PCFG に対する標準的なビタビアルゴリズムを拡張したアルゴリズムを用いる。

まず終端記号 (和音) \mathbf{z} から、開始記号 S に向かって内側確率を以下のように反復計算する。

$$p_{n,n}^A = \lambda_A \max_{c \in \Sigma} \eta_{A \rightarrow c} p(\mathbf{x}_n | c)^{1/I_n} \quad (17)$$

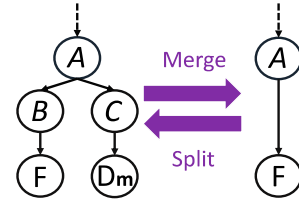


図 2: MH サンプラーにおける split-merge 操作

$$p_{n,n+k}^A = (1 - \lambda_A) \max_{B, C \in V} \theta_{A \rightarrow BC} p_{n,n+l-1}^B p_{n+l,n+k}^C \quad 1 \leq l \leq k$$

ここで、 $p(\mathbf{x}_n | c)$ は、音高系列の部分列 \mathbf{x}_n がコード c のもとで生成する確率であり、以下に与えられる。

$$p(\mathbf{x}_1 | c) = p(x_{1,1} | c) \prod_{i=2}^{I_1} p(x_{1,i} | x_{1,i-1}, c)$$

$$p(\mathbf{x}_n | c) = p(x_{n,1} | x_{n-1, I_{n-1}}, c) \prod_{i=2}^{I_n} p(x_{n,i} | x_{n,i-1}, c) \quad (n \geq 2) \quad (18)$$

そして、最尤の t と \mathbf{z} は、上述の計算におけるビタビパスを開始記号 S から再帰的にたどっていくことで得る。

4.3.2 Split-Merge サンプリング

MH 法の枠組みから、現在のサンプルから一つのコードを分割した、または隣り合うコードをマージしたサンプルを生成する手法 (Split-Merge サンプリング) を説明する。

split 操作の場合、新しいサンプル s^* は、 \mathbf{z} からコード z_n を確率的に選択し、 z_n の上にある非終端記号 $t_{n:n}$ を二つのノード t^L, t^R に分割し、 t^L, t^R からそれぞれ z^L, z^R を確率的に生成し、コード z^R のオンセットビート位置 $\phi^* \in (\phi_n, \phi_{n+1}) = [\phi_n + 1, \phi_{n+1} - 1]$ を確率的に生成する。したがって、提案分布は以下のように与えられる。

$$p(s^* | s) = \begin{cases} \frac{\theta_{t_{n:n} \rightarrow t^L t^R} \eta_{t^L \rightarrow z^L} \eta_{t^R \rightarrow z^R}}{N(\phi_{n+1} - \phi_n - 1)}, & \phi_{n+1} \geq \phi_n + 1; \\ 0, & \text{otherwise.} \end{cases} \quad (19)$$

一方、この操作に対する逆操作の提案分布 $p(s | s^*)$ は、merge 操作の提案分布に等しく、サンプル s はサンプル s^* において隣り合うコードの組 z^L, z^R を確率的に選択し、確率 $\eta_{t_{n:n} \rightarrow z_n}$ をもとにマージ後のコード z_n をサンプルする。したがって、

$$p(s | s^*) = \frac{\eta_{t_{n:n} \rightarrow z_n}}{\#\text{MergeableNodes}(s^*)} \quad (20)$$

が得られる。ここで、 $\#\text{MergeableNodes}(s^*)$ は、サンプル s^* において隣接するコードの組のうち、それぞれの葉ノードが木 t 内で部分木をなすものの数を表す。

サンプル s, s^* の尤度比は以下のように与えられる。

$$\frac{p(s^*)}{p(s)} = \frac{(1 - \lambda_{t_{n:n}}) \theta_{t_{n:n} \rightarrow t^L t^R} \lambda_{t^L} \eta_{t^L \rightarrow z^L} \lambda_{t^R} \eta_{t^R \rightarrow z^R}}{\lambda_{t_{n:n}} \eta_{t_{n:n} \rightarrow z_n}} \cdot \frac{p(\mathbf{x}^L | z^L) p(\mathbf{x}^R | z^R) p(\phi^* | \phi_n) p(\phi_{n+1} | \phi^*)}{p(\mathbf{x}_n | z_n) p(\phi_{n+1} | \phi_n)} \quad (21)$$

ここで、 \mathbf{x}^L , \mathbf{x}^R は、コード z_n をビート時刻 ϕ^* で分割したことによって分割された音高部分列を表す。式 (19), (20), (21) を用いて、式 (16) による s^* の受理率を算出する。

一方、merge 操作の場合、新しいサンプル s^* は、split 操作と似た方法で得られる。具体的には、merge 操作に対する受理率を式 (16) における s と s^* を交換すればよい。

4.4 コードのリズムの更新

ここでは、コードのリズム ϕ を事後分布 $p(\phi|t, z, \mathbf{x}, \psi, \Theta)$ にしたがって局所的に更新する方法について述べる。新しいサンプル s^* は、コード z_n をランダムで選択し、そのオンセットビート位置 ϕ_n を新しい位置 $\phi_n^* \in (\phi_{n-1}, \phi_{n+1})$ に移動して得られる。したがって、その提案分布 $p(s^*|s)$ とその逆操作の提案分布 $p(s|s^*)$ は以下のように得られる。

$$p(s^*|s) = p(s|s^*) = \frac{1}{N-1} \frac{1}{\phi_{n+1} - \phi_{n-1} - 1} \quad (22)$$

サンプル s , s^* の尤度比は以下のように与えられる。

$$\frac{p(s^*)}{p(s)} = \frac{p(\mathbf{x}_{n-1}^*|z_{n-1})p(\mathbf{x}_n^*|z_n)p(\phi_n^*|\phi_{n-1})q(\phi_{n+1}|\phi_n^*)}{p(\mathbf{x}_{n-1}|z_{n-1})p(\mathbf{x}_n|z_n)p(\phi_n|\phi_{n-1})p(\phi_{n+1}|\phi_n)} \quad (23)$$

ここで、 \mathbf{x}_{n-1}^* , \mathbf{x}_n^* は、コード z_n のオンセット位置 ϕ_n の更新によって、更新された音高部分列を表す。式 (22), (23) を用いて、式 (16) による s^* の受理率を算出する。

5. 評価実験

本章では、本稿の提案モデルと提案手法のそれぞれについて定量評価実験を行った。また、提案手法によって生成された和音系列の例を示した。

5.1 実験条件

和音系列生成モデル (PCFG) の学習には、The SALAMI Annotation Data [26] 内のポピュラー音楽 468 曲から抽出した、verse, bridge, chorus などのセクションに対応する 8 小節分の和音系列 705 個を用いた。コード記号の語彙は、ルート音 {C, C#, D, ..., B} とコードの種類 {major, minor} の組み合わせと、それ以外の “other” の 25 個とした。マルコフモデルの教師あり学習には、Rock Corpus [27] 内のポピュラー音楽 194 曲から抽出した、和音系列とメロディのペア 9902 組を用いた。以上の学習におけるすべてのハイパーパラメータの値は 0.1 とした。

メロディのテストデータには、RWC データベース [28, 29] 内のポピュラー音楽 69 曲から抽出した、セクションに対応する 8 小節のメロディ 290 個を用いた。なお、音楽コーパス内の楽曲の調はすべて C に移調した。

5.2 メロディに対する予測性能評価

PCFG に基づく階層的生成モデルのメロディに対する予測性能を評価するために、テストデータのメロディに対する学習パラメータの周辺尤度を算出する実験を行った。PCFG の非終端記号数 K は 1 以上 20 以下とした。各 K について、ギブスサンプリングを用いて異なるパラメータ θ , η , λ を推定し、それぞれのパラメータ集合に対して周辺尤度を算出した。パラメータ

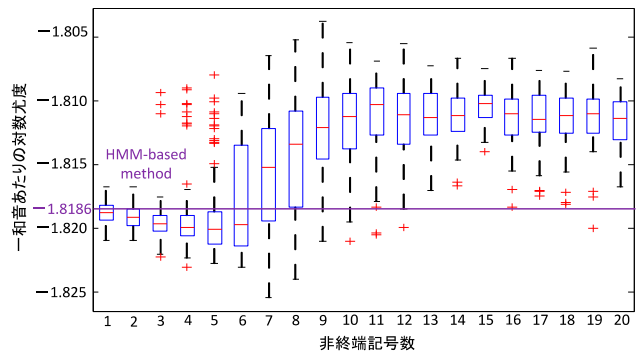


図 3: 一音符あたりの周辺尤度。箱ひげ図内で、赤線は中央値、赤色の十字は外れ値を示す。

集合数は、計算量の問題から 47 以上 100 以下とした。ただし、コードのリズムに関する拍節マルコフモデルを含むモデル全体を解析的に周辺化できないため、コードのオンセットは小節線に一致すると仮定した。また、ベースラインとして、16 分音符単位または小節単位でのコード記号の遷移を学習した HMM に基づくモデルをテストした。音符単位の場合には、音符はその時点のコード記号にしたがって生成されると仮定し、小節単位の場合には、音符系列は、式 (4) に示したマルコフモデルにしたがって生成されると仮定した。

未知のメロディ \mathbf{x} に対するパラメータ Θ の周辺尤度は、3.2 章における内側アルゴリズムに基づいて算出される。ただし、潜在木 t に加えて潜在和音系列 z についても周辺化するため、式 (11) の $p_{i,i}^A$ の算出を以下のように置き換える。

$$p_{i,i}^A = \lambda_A \sum_{c \in \Sigma} \eta_{A \rightarrow c} p(\mathbf{x}_i|c) \quad (24)$$

ここで、 $p(\mathbf{x}_i|c)$ は、式 (18) によって求められる。したがって、一音符あたりの平均周辺尤度 \mathcal{L} は以下に与えられる。

$$\mathcal{L} = \frac{1}{I} \log p(\mathbf{x}|\psi, \Theta) = \frac{1}{I} \log p_{0,N-1}^S \quad (25)$$

ここで、 I , N はそれぞれ音符、コードの数である。

実験結果を図 3 に示した。提案モデルは、HMM に基づく手法における最小単位が一音符の場合 ($\mathcal{L} = -3.2956$)、最小単位が一小節の場合 ($\mathcal{L} = -1.8186$) のいずれに対しても、その結果を上回った。周辺尤度は、非終端記号数 $K = 1$ から $K = 5$ までは微減し、 $K = 5$ を越えてからは増加する傾向にあった。

5.3 和音系列に対する予測性能評価

本稿で提案した和声付け手法が、未知のメロディに対して、和音系列をどれだけ予測できるかを評価するために、RWC データベースから抽出したメロディに対して和音系列を生成し、同データベースのコード進行アノテーションを正解データとして、16 分音符単位での一致率を求めた。PCFG の非終端記号数 K は 1 以上 20 以下とした。また、ベースラインとして、16 分音符単位でのコード記号の遷移を表現した HMM に基づく手法をテストした。

実験結果を図 4 に示した。提案モデルは、ベースライン手法における一致率 (15.12 %) を大きく上回った。非終端記号数

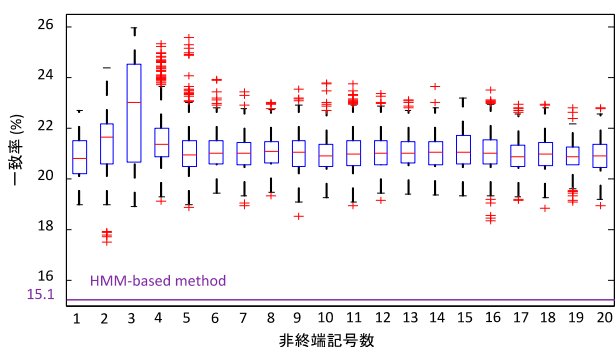


図 4: 一致率 [%]. 図 3 と同様の記法を用いる.

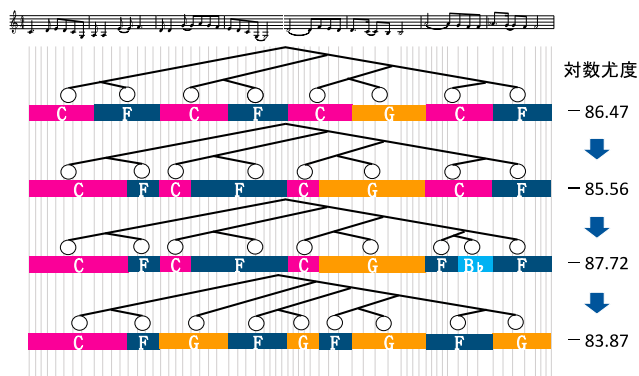


図 5: 入力メロディに対する最尤の和音系列探索の例.

$K = 3, 4, 5$ の場合、ベースラインモデルに対して特に高い一致率 (25 % など) を示していたが、 K と一致率の中央値には一般的な相関は見られなかった。

5.4 和音系列の生成例

提案手法の MH サンプリングによる最適な和音系列の探索例を図 5 に示した。PCFG の非終端記号数 K は 20 とした。図中の一歩上の和音系列は、コードのオンセット位置を小節線に固定した状態で、ビタビアルゴリズムによって初期化した状態である。二つ目の和音系列は、数回のサンプルを経て、2, 4, 6 個目のコードのオンセット位置が移動した状態である。三つ目の和音系列は、前の状態から 7 個目のコードが分割された状態である。一番下の和音系列は、入力メロディに対して最尤なサンプルを示す。この結果から、提案手法が和音系列の背後にある木構造を考慮しながら、可変長の和音系列を生成できていることが示された。

本手法には、今後取り組むべき問題も存在する。まず、提案手法は上述の例のように、単純な和音系列 (C, F など) を生成することが多い。これは和音系列がビタビアルゴリズムを用いて更新されるためである。一方、生成されたコードのリズムに関しても、二分音符にコードのオンセットが配置されるような単純なリズムを持つものが多く生成されていた。これは、split 操作が受理される回数が他の操作に対して少ないためであることが考えられる。

6. おわりに

本稿では、ポピュラー音楽コーパス内の音楽シンボルの背後

に潜む音楽規則の学習に基づいて、入力メロディに対する可変長の和音系列を生成することを目的とした自動和声付け手法を提案した。評価実験より、提案モデル・手法の HMM に基づく手法より優れた予測性能と、統計的作曲・編曲における実用性が示された。

ただし、提案手法は統計的学習に基づいているため、比較的単純な生成結果が得られる傾向がある。例えば、単純なリズムを持つ和音系列が多く生成されることに対しては、本手法の MH サンプラーに対して split 操作がより起きるような改良が必要である。単純なコード記号が多く生成されるということに対しては、和音系列の更新に現在用いているビタビアルゴリズムの代わりに、適切なサンプリング手法やビームサーチ手法を用いることが効果的である。

また、提案手法に対してはさらなる考察が必要である。本稿では、メロディは 4 分の 4 拍子であり、音符の最小単位は 16 分音符と仮定したため、最適な最小単位に関する考察や、他の拍子を適切に扱うためのモデル拡張が必要である。さらに、生成された和音系列の音楽的な適切さを評価するために、音楽理論に基づく定量的な評価実験や聴取実験が必要である。

謝辞: 本研究の一部は、JSPS 科研費 26700020, 16H01744 および JST ACCEL No. JPMJAC1602 の支援を受けた。

文 献

- [1] I. Simon, D. Morris, and S. Basu, “MySong: automatic accompaniment generation for vocal melodies,” Proceedings of the SIGCHI Conference on Human Factors in Computing Systems ACM, pp.725–734 2008.
- [2] S.A. Raczynski, S. Fukayama, and E. Vincent, “Melody harmonization with interpolated probabilistic models,” Journal of New Music Research, vol.42, no.3, pp.223–235, 2013.
- [3] H. Riemann, Harmony Simplified: or, The Theory of the Tonal Functions of Chords, Augener, 1896.
- [4] W. Maler, Beitrag zur Durmolltonalen Harmonielehre I (7th ed.), F. E. C. Leuckart, 2007.
- [5] M.J. Steedman, “A generative grammar for jazz chord sequence,” Music Perception, vol.2, no.1, pp.52–77, 1984.
- [6] C.H. Chuan and E. Chew, “A hybrid system for automatic generation of style-specific accompaniment,” Proceedings of the 4th International Joint Workshop on Computational Creativity, pp.57–64, 2007.
- [7] K. Ebcioglu, “An expert system for harmonizing four-part chorales,” Computer Music Journal, vol.12, no.3, pp.43–51, 1988.
- [8] G. Papadopoulos and G. Wiggins, “AI methods for algorithmic composition: A survey, a critical view and future prospects,” AISB Symposium on Musical Creativity, pp.110–117, 1999.
- [9] M. Towsey, A. Brown, S. Wright, and J. Diederich, “Towards melodic extension using genetic algorithms,” Educational Technology & Society, vol.4, no.2, pp.54–65, 2001.
- [10] R.D. Prisco and R. Zaccagnino, “An evolutionary music composer algorithm for bass harmonization,” Applications of Evolutionary Computing, pp.567–572, 2009.
- [11] M. Allan and C. Williams, “Harmonising chorales by probabilistic inference,” Advances in Neural Information Processing Systems (NIPS), pp.25–32, 2005.
- [12] R. Groves, “Automatic harmonization using a hidden semi-Markov model,” Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference (Boston, MA), pp.48–54, 2013.
- [13] J.F. Paiement, D. Eck, and S. Bengio, “Probabilistic

- melodic harmonization,” Conference of the Canadian Society for Computational Studies of Intelligence, pp.218–229, 2006.
- [14] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*, MIT press, 1985.
- [15] M. Hamanaka, K. Hirata, and S. Tojo, “Implementing ‘A generative theory of tonal music’,” *Journal of New Music Research*, vol.35, no.4, pp.249–277, 2006.
- [16] M. Hamanaka, K. Hirata, and S. Tojo, “Musical structural analysis database based on GTTM,” *ISMIR*, pp.325–330, 2014.
- [17] E. Nakamura, M. Hamanaka, K. Hirata, and K. Yoshii, “Tree-structured probabilistic model of monophonic written music based on the generative theory of tonal music,” *IEEE ICASSP*, pp.276–280, 2016.
- [18] R. Scholz, E. Vincent, and F. Bimbot, “Robust modeling of musical chord sequences using probabilistic N-grams,” *IEEE ICASSP*, pp.53–56, 2009.
- [19] K. Yoshii and M. Goto, “A vocabulary-free infinity-gram model for nonparametric bayesian chord progression analysis,” *ISMIR*, pp.645–650, 2011.
- [20] J.F. Paiement, D. Eck, and S. Bengio, “A probabilistic model for chord progressions,” *ISMIR*, pp.312–319, 2005.
- [21] M. Rohrmeier, “Mathematical and computational approaches to music theory, analysis, composition and performance,” *Journal of Mathematics and Music*, vol.5, no.1, pp.35–53, 2011.
- [22] D. Quick, “Learning production probabilities for musical grammars,” *Journal of New Music Research*, vol.45, no.4, pp.295–313, 2016.
- [23] S. Fukayama, K. Yoshii, and M. Goto, “Chord-sequence-factory: A chord arrangement system modifying factorized chord sequence probabilities,” *ISMIR*, pp.457–462, 2013.
- [24] S.F. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” *Association for Computational Linguistics (ACL)*, pp.310–318, 1996.
- [25] M. Johnson, T.L. Griffiths, and S. Goldwater, “Bayesian inference for PCFGs via Markov chain Monte Carlo,” *North American Chapter of the Association for Computational Linguistics - Human Language Technologies (NAACL-HLT)*, pp.139–146, 2007.
- [26] J.B.L. Smith, J.A. Burgoyne, I. Fujinaga, D.D. Roure, and J.S. Downie, “Design and creation of a large-scale database of structural annotations,” *ISMIR*, pp.555–560, 2011.
- [27] T.D. Clercq and D. Temperley, “A corpus analysis of rock harmony,” *Popular Music*, vol.30, no.01, pp.47–70, 2011.
- [28] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, “RWC music database: Popular, classical and jazz music databases,” *ISMIR*, vol.2, pp.287–288, 2002.
- [29] M. Goto, “Aist annotation for the RWC music database,” *ISMIR*, pp.359–360, 2006.