社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2008－MUS－74 (27)
2008－SLP－70 (27)
2008／2／9

マルコフ確率場を用いた自動作曲

Automatic Composition using Markov Random Fields

住田 浩之† 林 朗†
Sumida Hiroyuki Hayashi Akira

本論文では，マルコフ確率場を用いた自動作曲を提案する．ある作曲家の曲を，マルコフ確率場の特徴関数と重みで表現することで特徴を学習し，この確率場から確率の高い標本をモンテカルロ法により得ることで自動作曲を行う．

In this paper, we propose an automatic composition method for polyphony music using Markov random fields. Given pieces of music by a composer. we learn the characteristics of the pieces in the form of feature functions and their weights for Markov random fields. Then, a new piece is automatically composed by obtaining a sample with high probability for the random field using a Monte Carlo method.

1 研究目的

近年，ウェブサイトコンテンツや映像作品に使用するBGMなど，任意のテーマに添い尚且つ著作権フリーな音楽素材への需要が高まってきている．このことを解決する一つの方法が，自動作曲によって作った曲を使用するというものである．自働作曲とは，曲調，テンポなどの任意のパラメータを与えることで，これに従い曲が生成されるというものである．これにより，使用用途に合わせたオリジナルの曲を作ることができる．

これまでの自動作曲の分野では，時刻についての1次元上で，前の時刻の音の配置やパターンから現在の音の出る確率を学習するというマルコフ連鎖を用いた手法が主流であった．しかしこの方法は多声を表現する場合，多声音数の累乗の計算量が必要であり，またトリル等の音楽表現のように同じフレーズが極端に多く学習されるとそのフレーズの繰り返しに陥りやすい，といった問題があった．

そこで本論文では，マルコフ確率場を用いた自動作曲を提案する．マルコフ確率場とは2次元画像処理などに用いられる技術であるが，学習データをある作曲家のMIDIデータ（縦に音階，横に時刻とし2次元の格子として扱う）とし，作曲家のスタイルや特徴をマルコフ確率場の特徴関数と重みで表現することで画像と同様に特徴の学習を行い，この確率場から確率の高い標本をモンテカルロ法により得ることで自動作曲を行ってゆく．

2 手法

2.1 学習データの表現

本研究であつかう学習データはMIDI形式の楽譜データとし，以下の手順で変換する．

1. 音を1オクターブ上にまとめる．
2. 音のリズム情報をなくし，休符は詰める．
3. 音の状態を $\{1(off), 2(on), 3(begin)\}$ とし，$T=1$ に音の開始を表現する3を挿入する．

縦軸に音階 $i=1...12$, 横軸に時間 $t=1...T$ であるような $12*T$ の格子を作る．

† 広島市立大学大学院 情報科学研究科 知能工学専攻 パターン認識研究室 〒731-3194 広島市安佐南区大塚東3-4-1
Email: sumida@robotics.im.hiroshima-cu.ac.jp

2.2 確率場の表現

2.2.1 マルコフ確率場とは

2次元配置空間 Ω における各格子 $(i,j) \in \Omega$ のそれぞれに確率変数 $x_{i,j}$ が割り当てられているとき，$\mathbf{x} = \{x_{i,j} | (i,j) \in \Omega\}$ を確率場と呼ぶ．このとき，$p(\mathbf{x}) = \sum_{\mathbf{x}} p(x) = 1.0$ である．

$\mathbf{x} = \{x_{i,j}\}$ は通常空間的な依存性があり，Ω で定義された近傍系 $\{\mathcal{N}_{i,j} | (i,j) \in \Omega\}$ に対して以下のマルコフ性を満たすとき，この確率場をマルコフ確率場という．

$$p(x_{i,j} | x_{l,m}, (l,m) \neq (i,j)) = p(x_{i,j} | x_{l,m}, (l,m) \in \mathcal{N}_{i,j}) \tag{1}$$

(1) 式は，近傍における確率変数 $\{x_{l,m} | (l,m) \in \mathcal{N}_{i,j}\}$ が与えられれば，$x_{i,j}$ は非近傍における確率変数からは独立であることを示している．

2.2.2 マルコフ確率場の適用

学習データを $12 \times T$ の格子とする．時刻 t，音階 i に位置する音 $n_{i,t}$ を**音素**とし，各音素 $n_{i,t}$ を確率変数とみなしたマルコフ確率場 $\mathbf{n} = \{n_{i,t} | 1 \leq i \leq 12, 1 \leq t \leq T\}$ を考える．マルコフ確率場の近傍系を $\{\mathcal{N}_{i,t} | 1 \leq i \leq 12, 1 \leq t \leq T\}$ としたとき，(1) 式は，近傍 $\mathcal{N}_{i,t}$ における音素が与えられれば，音素 $n_{i,t}$ は非近傍における音素からは独立であることを示す．

2.2.3 特徴関数

近傍内の音の配置を**特徴**とし，特徴関数で表現する．特徴関数とは基準の音となる音素 $n_{i,t}$ と，その近傍 $\mathcal{N}_{i,t}$ の音 $N_{i,t} = \{n_{j,t+\Delta t} | (j, t+\Delta t) \in \mathcal{N}_{i,t}\}$ の配置が与えられたパターンと一致するときに 1 を返す関数である．特徴関数は以下の形で表現される

$$f_k(n_{i,t}, N_{i,t}) = \delta(i = i^{(k)}) \delta(n_{i,t} = n^{(k)}) \times \prod_{(j,\Delta t,n) \in S_k} \delta(n_{j,t+\Delta t} = n) \tag{2}$$

ただし，$S_k = \{(j, \Delta t, n) | \Delta t = \text{整数}, 1 \leq j \leq 12, 1 \leq n \leq 3\}$ を位置 (音階 j，時刻差 Δt) とその位置の音の状態 $n (1 \leq n \leq 3)$ の対の集合とし，$\delta(a = b) = 1(a = b), 0(otherwize)$ をデルタ関数とする．

逆に，上記のように特徴関数の集合 $\{f_k | 1 \leq k \leq K\}$ が与えられた時，近傍 $\mathcal{N}_{i,t}$ は $\{S_k | 1 \leq k \leq K\}$ により以下のように定まる．

$$\mathcal{N}_{i,t} = \{(j, t+\Delta t) | 1 \leq^{\exists} k \leq K, i = i^{(k)}, (j, \Delta t, n) \in S_k\} \tag{3}$$

表 1 に特徴関数の例を示す．1 列目が Δt(近傍の相対的時刻) を，2 列目が音階 1...12 を，3 列目が状態 $\{1, 2, 3\}$ を表している．また，2 行以上の特徴は，複数の音から構成される特徴を表している．

- (A) は “ソ (音階 8)” が出る (状態 2) であるという特徴
- (B) は “ファ” と “ラ” が同時に出た後に “ソ” が出るという特徴
- (C) は “ミ” の音で始まるという特徴

(A)

-	8	2

(B)

-	6	2
0	10	2
1	8	2

(C)

-	5	2
-1	0	3

表 1　特徴関数の例

2.2.4 ギブス分布

各特徴関数 f_k にそれぞれ重み λ_k が割り当てられているとき，確率場を特徴関数と重みの積の和をポテンシャルとするギブス分布の形であらわすことができる．

$$\hat{P}(\mathbf{n}) = \frac{1}{Z_{\hat{P}}} \exp \sum_{t=1}^{T} \sum_{i=1}^{12} \sum_{k=1}^{K} \lambda_k f_k(n_{i,t}, N_{i,t}). \tag{4}$$
$$\mathbf{n} = \{n_{1,1}, n_{2,1}, ..., n_{12,T}\}$$

ここで，$Z_{\hat{P}}$ は正規化定数で以下の形となる．

$$Z_{\hat{P}} = \sum_{n=1}^{3} \exp \sum_{t=1}^{T} \sum_{i=1}^{12} \sum_{k=1}^{K} \lambda_k f_k(n, N_{i,t}) \tag{5}$$

2.2.5 確率場の学習

$\mathbf{n}$ を学習データとし，確率場の尤度 $\hat{P}$ を最も大きくするような特徴集合 $\{\lambda_k, f_k | k = 1 \leq k \leq K\}$ を導きたい．しかし，膨大な特徴の組み合わせの中から最も大きな尤度を得るような組み合わせを見つけ出すのは計算量的に困難である．そこで，欲張りアルゴリズム (Greedy Algorithms) を用いる．欲張りアルゴリズムとは，各ループにおいて最も尤度を大きくするような特徴を取り入れていきながらループを繰り返すことで，計算量を抑えながらも尤度が大きくなるような特徴の組み合わせを得ることができるというものである．

2.2.6 対数疑似尤度

最も確率場を改善する特徴を見つけ出すために，最尤推定を行う．しかし，(4) 式を計算するとき，正規化定数である $Z_{\hat{p}}$ の計算は $\{n_{i,t} | 1 \leq i \leq 12, 1 \leq t \leq T\}$ の組み合わせの数 (3^{12T}) だけ行わねばならず，莫大な計算量になってしまう．

そこで，**近傍の依存関係のみに注目した対数疑似尤度を用い対数尤度を近似**する．対数疑似尤度は以下の形となる．

$$\begin{aligned} \mathcal{PL}_{\hat{p}} &= \frac{1}{12T} \log \prod_{t=1}^{T} \prod_{i=1}^{12} \hat{P}(n_{i,t} | N_{i,t}) \\ &= \frac{1}{12T} \sum_{t=1}^{T} \sum_{i=1}^{12} \log \hat{P}(n_{i,t} | N_{i,t}) \end{aligned} \tag{6}$$

$$\hat{P}(n_{i,t} | N_{i,t}) = \frac{1}{Z_{i,t}} \exp \sum_{k=1}^{K} \lambda_k f_k(n_{i,t}, N_{i,t}). \tag{7}$$

ここで，$Z_{i,t}$ は正規化定数で以下の形となる．

$$Z_{i,t} = \sum_{n=1}^{3} \exp \sum_{k=1}^{K} \lambda_k f_k(n, N_{i,t}). \tag{8}$$

2.3 特徴選択

前節で触れた欲張りアルゴリズムを用いた特徴選択アルゴリズムの概要は以下の 4 つのステップからなる．

1. 特徴選択の候補となる**候補特徴集合**を作成する．
2. 作成した候補特徴のそれぞれの**対数疑似尤度のゲイン (対数疑似尤度の改善量) を計算**する．
3. **最も高い改善量の候補特徴を選択**し，特徴集合 $\mathcal{F}$ に加える．
4. 新しい特徴集合での**各重みを再計算**する．

以上のループを任意の特徴数 K 個だけ特徴を得るまで行うことで特徴集合 $\mathcal{F}$ を得ることができる．

2.3.1 候補特徴集合の作成

候補特徴集合は**原始特徴**及び**候補特徴**の 2 種類から成る．それぞれの特徴は以下のように作られる．

- 原始特徴 単一の音のみで構成される (ドのみ等) 特徴であらかじめ与えられている．
- 候補特徴 特徴集合 $\mathcal{F}$ に含まれる既存の特徴に，$t = \pm 1$ の範囲で原始特徴を加えた特徴で，各ループの候補特徴作成ステップで作られる．

候補特徴集合を $\mathcal{G}$ とし，各候補特徴を $g_h \in \mathcal{G}$ とする．なお，$h \in H$ は各候補特徴のインデックスである．

2.3.2 各候補のゲインを計算

対数疑似尤度のゲインは以下のように定める．各候補特徴を加えた確率場の対数疑似尤度 $\mathcal{PL}_{\hat{p}+\{\lambda_h^g\}}$ と，加える前の確率場の対数疑似尤度 $\mathcal{PL}_{\hat{p}}$ との差をゲインの式 (9) とし，各特徴関数 g についてそれぞれゲインを計算する．

$$Gain(g_h, \lambda_h^g) = \mathcal{L}_{\hat{p}+\{\lambda_h^g \cdot g_h\}} - \mathcal{L}_{\hat{p}} = \lambda_h^g \tilde{E}[g_h] - \log \hat{E}[\exp(\lambda_h^g \cdot g_h)] \tag{9}$$

ただし，λ_h^g を各候補特徴 g_h の重みとする．また，$\tilde{E}[g_h]$ は特徴 g_h の経験分布に関する期待値，$\hat{E}[g_h]$ はモデル分布に関する期待値であり，以下のようになる．

$$\tilde{E}[g_h] = \frac{1}{12T} \sum_{t=1}^{T} \sum_{i=1}^{12} g_h(n_{i,t}, N_{i,t}) \tag{10}$$

$$\hat{E}[g_h] = \frac{1}{12T} \sum_{t=1}^{T} \sum_{i=1}^{12} \sum_{n=1}^{3} g_h(n, N_{i,t}) \hat{P}(n|N_{i,t}) \tag{11}$$

2.3.3 候補特徴の選択

各候補特徴について，式 (9) の λ_h^g に関する偏微分の式 (12) からゲインを最大にする重み λ_h^g を計算する．

$$\frac{\partial}{\partial \lambda_h^g} Gain(g_h, \lambda_h^g) = \tilde{E}[g_h] - \frac{\hat{E}[g_h \exp(\lambda_h^g g_h)]}{\hat{E}[\exp(\lambda_h^g g_h)]} = 0 \tag{12}$$

λ_h^g に関する勾配が 0 となるときに各特徴は最大のゲインをとり，これを各特徴の対数疑似尤度の改善量とする．この中から，対数疑似尤度を改善する，つまり最も高いゲインを持つ候補特徴を特徴集合 $\mathcal{F}$ へ，またその時の重みを重み集合 Λ に加える．

2.3.4 重みの再計算

2.3.1 での特徴選択は，候補以外の特徴の重みは固定させたまま行った．よって，新しくなった特徴集合における各重みを再計算する必要がある．各特徴の重みを，対数疑似尤度の式 (6) を最大にするように計算していく．この最大化に必要な (6) 式の偏微分は以下の式で表される．

$$\frac{\partial}{\partial \lambda_k} \mathcal{PL}_{\hat{P}} = \tilde{E}[f_k] - \hat{E}[f_k] \tag{13}$$

式 (13) の勾配が 0，すなわち特徴 f_k の経験分布に関する期待値と，モデル分布に関する期待値の差が 0 となるように，重み $\lambda_k (1 \leq k \leq K)$ を設定する．

2.3.5 特徴選択アルゴリズム

ここまでのアルゴリズムをまとめると以下のような手順になる．

- *Initial Data:*
  学習データ $\{n_{i,t} | 1 \leq i \leq 12, 1 \leq t \leq T\}$ 及び初期モデル $\hat{P}_0$
  学習する特徴の数 K

- *Output:*
  特徴集合 $\mathcal{F} = f_0, ..., f_K$
  各特徴の重み集合 $\Lambda = \lambda_0, ..., \lambda_K$
- *Algorithm:*
  (0) $\hat{P}^{(0)} = \hat{P}_0$.
  (1) (9) 式で各候補特徴 $g \in \mathcal{G}$ のゲインを計算する.
  (2) 最も高いゲインを得た内上位 1 %を候補特徴とし，特徴集合 f_n に加える.
  (3) 式 (13) で各特徴の重みを再計算し，またその時の確率場 $\hat{P}_*$ を計算する.
  (4) $\hat{P}^{(n+1)} = \hat{P}_*$ 及び $n \leftarrow n+1$ とし，任意の特徴数 K 個に達するまでステップ (1) に戻る.

2.4 自動作曲

ここまでで学習した特徴とその重みから，新しい曲の自動作曲を行う．学習パートで得られた特徴集合とその重みからシミュレーティッドアニーリングを用い，音を配置する手順をいかに示す.

1. $12 \times T$ の各格子にランダムに音を配置 $(1 or 2)$ する．また，$t = 1$ には曲の開始を示す 3 を挿入する.
2. $t \leq 2$ 以降の各格子のエネルギーを特徴とその重みから計算する.
3. $t \leq 2$ 各格子のそれぞれでに対してランダムに音を生成し，そのエネルギーと前ステップでの各格子のエネルギーとを比較し大きければ新しい音に差し替える．また，局所解に陥らないために，そうでなくてもある一定確率（ただし，温度変数 $Temp$ に依存する）で差し替える.
4. 温度変数 $Temp$ を少し下げステップ 2 に戻る．これを対数疑似尤度に目立った変化がなくなるまで繰り返す.

このアルゴリズムを十分回数行ったとき，出力される格子 $12 \times T$ が自動作曲された曲となる.

3 実験

バッハのピアノ組曲 〃フランス組曲 第 5 番 ト長調 *BWV 816*"から全 6 曲を学習データとして，以下の条件で自動作曲を行った．なお，学習された特徴や自働作曲された曲は以下のサイトで mp3 ファイルが入手できる.
http://heru3.com/m74/

- 特徴の学習数 $K = 1000$ とした
- 学習した特徴から $T = 100$ として自動作曲を行った

3.1 特徴選択

特徴選択アルゴリズムによって学習した特徴集合で重みの最も大きかった物から 10 個に注目すると，これらの音は不協和音が少なく，学習データで与えた曲の通り ト長調に出る音が多かった.

同様に，重みの小さかった 10 個に注目すると，これらの音は不協和音であったり，ト長調には出てこない音が多かった.

3.2 自働作曲

学習した特徴集合を用い，自働作曲アルゴリズムによって自働作曲を行い，図 1 のような曲が出力された.

3.3 実験考察

多少の不協和音はみられるものの，大きい重みの特徴の音は出現やすく，小さい重みの特徴の音は出現しにくい．また，重みが負の値をとった特徴は曲中に出てきていない.

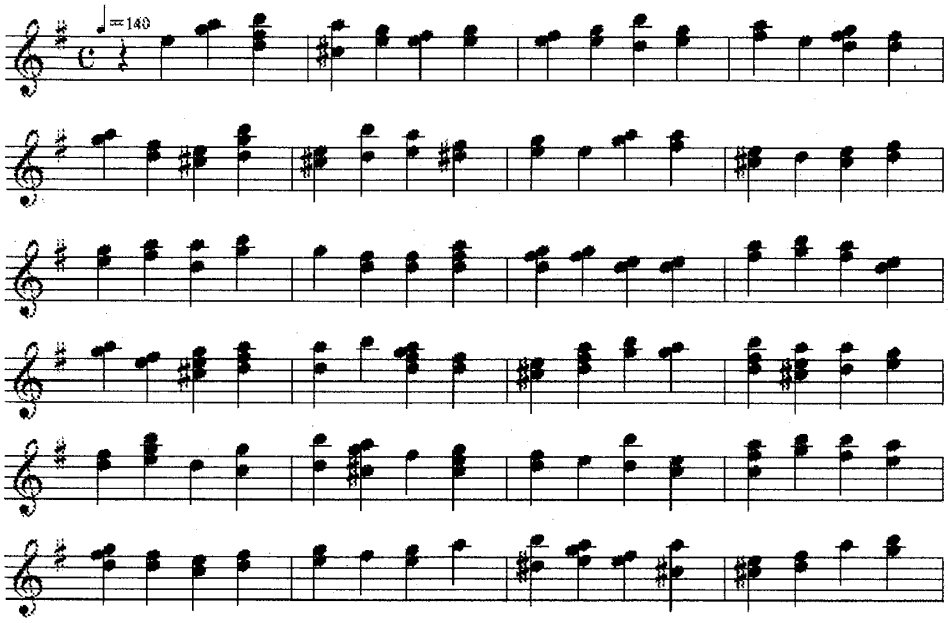

図1　自働作曲アルゴリズムによって作られた曲

4　まとめと今後の課題

特徴学習を用いた自動作曲というアプローチで，新しい曲を作ることができた．今後の課題として

- バッハ以外の作曲家についても実験を行う．
- リズムも特徴として学習し，自動作曲に取り入れる．
- 1オクターブに限定せずに学習，作曲を行う．
- このアルゴリズムでは音楽的な知識なしで機械的に音を配置しているにすぎないので，事前知識として音楽の基本的な知識を与える．

などが考えられる．

参考文献

[1] Stephen Della Pietra, Vincent Della Pietra, and John Lafferty, Member, IEEE, “Inducing Features of Random Fields” IEEE TRANSACTIONS PATTERN ANALYSIS AND MACHINE INTELLIGENCE,VOL. 19, NO. 4, APRIL 1997

[2] Victor Lavrenko, Jeremy Pickens, “Polyphonic Music Modeling with Random Fields” Center for Intelligent Information Retrieval Department of Computer Science University of Massachusetts, Amherst, MA 01003 USA

[3] S. Z. Li, “Markov random field modeling in computer vision” Springer Verlag, 1995.