

対話型進化的計算手法による作曲システムにおける HMM の作曲法*

蓮井洋志† 小倉久和‡

室蘭工業大学情報工学科§ 福井大学大学院工学研究科知能システム工学専攻¶

1 はじめに

我々は、対話型進化的計算手法を用いて作曲するシステム *i-Sonneteer* を作成している。このシステムは HMM を用いてユーザの好みのモデルをシステム内に作ることを目標とする。逐次的リズム音程生成モデル [1] によるリズム生成方法を参考に、音高の生成もできるように応用した HMM を考案した。この HMM は決定論的に学習し、メロディーを生成する。進化的計算手法では、HMM を個体とし、生成したメロディーの評価を HMM の適応度とする。評価はユーザが行ない、ユーザが GUI によってメロディーを好みの形に編集する。編集後の音楽をうまく選択して学習すれば、ユーザの好みのモデルに最適化できる。

HMM は逐次的リズム音程生成モデルではランダムウォークで生成していたが、(1) 同じ HMM が同じメロディーを生成しないこと、(2) 生成したメロディーの質が悪いことから、1 音符当たりの確率が一番高いメロディーを生成するように変更した。本稿では、生成方法と、生成したメロディーの評価実験について述べる。

2 HMM の学習、生成と最適化

2.1 楽譜の k -measure HMM での学習

HMM は状態と記号からなる記号列学習モデルである。メロディーの音休符を記号として考え、音休符列を学習する。

k -measure HMM の学習方法を提案する。 k 小節ごとにメロディーの学習、生成を行なう。

音符の属性としては、開始半拍数、音高、音符の長さの 3 つがある。開始半拍数 i は $(k \times l + 1)$ 小節目の最初からの半拍数とし、音高を p 、音符の長さは d と記述する。 d の単位は半拍である。例えば、音高 p が 4 オクターブ目の C である場合、 $p = C4$ と書く。また、休符は $p = REST$ と記述する。音の強さは一定とする。例

*Composing Method of HMM in Composing System with Iterative Evolutionary Calculation

†Hiroshi Hasui

‡Hisakazu Ogura

§Department of Computer Science and Systems Engineering in Muroran Institute of Technology

¶Department of Human and Artificial Intelligent Systems, Fukui University

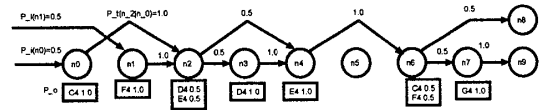


図 1: 学習した 1-measure HMM



図 2: 学習するメロディーの楽譜

えば、 $k = 2$ の場合は、開始拍数が 3 小節目の 3 拍目で音高が 4 オクターブ目の D の 4 分音符は、 $l = 1$ 、 $i = 4$ 、 $p = D4$ 、 $d = 2$ である。

開始半拍数を i としたとき、状態のラベルを n_i とする。この状態が n_i から n_j に遷移した場合、音符の長さは $d = j - i$ で、この状態の記号出力は音高 p である。

学習は、HMM の状態遷移確率などを適応化することによって行なう。学習手順を簡単に説明する。音符の開始半拍数が i 、音高が p 、音符の長さが d の場合、状態は n_i から n_{i+d} に遷移する。また、状態 n_i のときの記号は p である。 $i + d > 8 \times k$ の場合は、 $n_{8 \times k + 1}$ に遷移する。メロディーの全ての音休符に対して、各状態から各状態への遷移した音符の数、各状態における各記号の音符の数を数える。 k -measure HMM の学習パラメータは以下のように算出する。

$$P_i(i) = \frac{(k \times l + 1) \text{ 小節目の最初の音符の状態が } n_i \text{ の数}}{\text{学習する全小節数を } k \text{ で割った数}} \quad (1)$$

$$P_i(j|i) = \frac{n_i \text{ から } n_j \text{ に遷移する音休符の数}}{n_i \text{ の音休符の数}} \quad (2)$$

$$P_o(o|i) = \frac{n_i \text{ の記号が } o \text{ である音休符の数}}{n_i \text{ の音休符の数}} \quad (3)$$

図 2 の楽譜を 1-measure HMM で学習する様子を説明する。音符 j は楽譜の最初から見て j 番目の音符を示す。音符 1 は $l = 0$ 、 $i = 0$ 、 $p = C4$ 、 $d = 2$ である。この小節の最初の音符であるために初期状態が n_0 となる。

音符 1 は状態はノード n_0 から n_2 に遷移する。状態 n_0 にときの記号は $C4$ である。音符 2 は $l = 0$ 、 $i = 2$ 、 $p = D4$ 、 $d = 2$ である。状態はノード n_2 から n_4 に遷移する。状態 n_2 にときの記号は $D4$ である。音符 3、4 は



図 3: 1-measure HMM が生成したメロディー

同様に学習する。音符 5 は次の小節にタイでつながる音符である。この音符の場合は n_9 に遷移する。 n_9 は次の小節にタイでつながる音符の遷移するノードである。状態が n_7 から n_9 に遷移し、状態 n_7 の記号は $G4$ となる。

音符 6 は、2 小節目の第 1.5 拍で、各パラメータは $l = 1$ 、 $i = 1$ 、 $p = F4$ 、 $d = 1$ である。この小節の最初の音符であるために初期状態が n_1 となる。初期状態が n_0 と n_1 の 2 つになった。両方の状態とも 1 音符だけしか存在しないので初期状態確率は、両方とも 0.5 となる。

音符 6 は状態はノード n_1 から n_2 に遷移する。状態 n_1 の記号は $F4$ である。音符 7 は $l = 1$ 、 $i = 2$ 、 $p = E4$ 、 $d = 1$ で n_2 から n_3 に遷移する。 n_2 から遷移する音符は音符 2 と音符 7 があり、それぞれ状態 n_4 、 n_3 に遷移する。各々の状態への遷移確率は 0.5 となる。記号も $D4$ と $E4$ の 2 つがあり両方とも記号出力確率は 0.5 となる。音符 11 は小節のくぎりで終るために n_8 に遷移する。 n_8 、 n_9 は出力記号を持たない。

図 2 の楽譜を 1-measure HMM が学習した結果、図 1 のようになる。長方形に囲まれた数字が記号出力確率、2 つのノードを結ぶ矢印の上の数字が状態遷移確率である。初期状態確率は左端の矢印の上の数字である。

2.2 k -measure HMM によるメロディーの生成

k -measure HMM は学習したデータに基づいて、 k 小節のメロディーを生成する。 k -measure HMM は 1 音符あたりの平均の状態遷移確率が一番高いリズムを探索する。探索には Dijkstra 法を用いた。各々の状態遷移確率の対数の逆数を取って $value$ を減算したものを距離とし、距離の和が一番小さいノード列を探索する。 n_0 から $n_{8 \times k}$ への最短距離を持つノード列がリズムを表す。

$value$ は、0 ではない状態の数を n 、0 ではない状態遷移確率を $p_{state,i}$ とすると以下の式で表される。

$$value = -\frac{\sum_i \log(p_{state,i})}{n} \quad (4)$$

音符の開始半拍数だけを考えた場合の 1 音符あたり生起確率の対数の逆数を $value$ とした。音符列のリズムの生起確率は状態遷移確率の積で計算されるために、すべての確率の積を計算する。確率が 0 の状態遷移には音符がないために生成される可能性がない。これは $value$ の

N	要修正曲数	評価の高い曲数	平均評価値
2	10	1	0.559
3	13	1	0.519
4	12	1	0.513
5	9	0	0.5355
6	6	0	0.549
7	5	3	0.6715
8	3	2	0.634

表 1: 修正する必要があると考えた曲数、評価の高い曲数と HMM の平均評価値

計算の対象外とした。距離の和を計算するとき、 $value$ を減算しないと、音符数の少ないメロディーが生成されてしまう。

音高は、音符の長さの列を決定したときの各状態における最大の記号出力確率を持つ記号を使った。もし同じ確率であれば、音高の低いものを使った。

図 3 に前節で学習した 1-measure HMM が生成したメロディーを示す。

3 実験と考察

被験者が自動作曲システムの作曲した曲を 20 曲修正し、その中から N 曲重複をゆるさずに 20 通りランダムな選択方法で音楽グループを作った。それらの音楽グループを 8-measure HMM に学習させた。その 8-measure HMM の生成したメロディーを被験者が自分の好みにどのくらいあっているかを 0~1 の値で評価した。被験者が修正する必要があると考えたメロディーの数と学習した曲以上の評価を得たメロディーの数、およびメロディーの平均評価値を表 1 に示す。修正することで良くなる可能性のある曲は必ず修正するものとする。被験者は音楽鑑賞を趣味にして 8 年になる情報工学の大学生である。

7 曲学習した HMM は、平均評価値が一番高いにも関わらず、修正の必要性がある曲が 8 曲学習した HMM より多かった。修正の必要のある曲は、修正によってより評価が高くなる可能性が高い。修正が可能であるこのシステムにおいては、7 曲学習するのが一番評価の高いメロディーを作ることができる。評価が高いということは、ユーザの好みを反映しているということである。それらを次の世代で HMM が学習すると、生成するメロディーに好みが反映されやすいと考える。そこで、本システムでは $N = 7$ が良いと思われる。

参考文献

- [1] 川村 修 大園 忠親 伊藤 孝行 新谷 虎松: 逐次のリズム音程生成モデルに基づく自動作曲, 情報処理学会音楽情報科学研究会, Vol. 63, pp. 19-24 (2005).