

隠れマルコフモデルを用いた多声音楽の楽譜追跡

Score following of polyphonic music with Hidden Markov Model

五十嵐 巧輝† 山田 昌尚†

Kouki IGARASHI, Masanao YAMADA

1.はじめに

楽譜追跡とは、演奏者が譜面上のどの部分を演奏しているかをリアルタイムに推定するものである。応用例として演奏の進行に合わせて自動で楽譜をめくる自動譜めくりや、演奏に伴奏をつける自動伴奏等の技術が上げられる[1]。一般的な楽譜追跡では、一つのパートもしくは音楽全体の楽譜追跡を扱っているが、多声音楽において個別のパートごとに楽譜追跡をすることは行われていない。一方、合奏の練習を行う際には楽譜を使用して練習することが想定される。合奏の際には、パートごとに発音タイミングを合わせることは重要である。しかし、合奏時にパートごとの小さな発音タイミングのずれを知覚することは困難である。さらに、楽譜上とは異なった音高を演奏してしまうことも想定される。

そこで、本研究では多声音楽を対象として隠れマルコフモデルを用い、個別の演奏パートごとに楽譜追跡を行い、発音タイミングのずれを定量的に表現する手法を提案する。本稿では、音高情報を使った多声音楽の楽譜追跡の手法と、楽譜追跡を行った結果について述べる。

2.楽譜追跡

隠れマルコフモデルは、観測された観測系列から直接知ることのできない状態の遷移を推定することができるモデルであり、一般的な楽譜追跡に用いられる。隠れマルコフモデルは、状態系列、状態遷移確率、観測系列、出力確率の4つの要素からなる。本研究では、多声音楽の楽譜追跡を行うため、Factorial HMM (階層隠れマルコフモデル) [2]を用いた。これは、1つの観測系列から複数の状態系列を扱うモデルである。本稿では状態系列を楽器の演奏パート数だけ用意し、1チャンネルのMIDIデータから複数の演奏パートの状態系列の遷移を推定する問題を扱う。状態系列数が2のモデルを図1に示す。 $z_n^{(k)}$ は状態系列 k の n 番目の要素、 x_n は n 番目の観測系列の要素、 $p(z_n^{(k)}|z_{n-1}^{(k)})$ は状態 $z_{n-1}^{(k)}$ から $z_n^{(k)}$ への遷移確率、 $p(x_n|z_n^{(k)})$ は状態 $z_n^{(k)}$ に対応する x_n の出力確率である。 $\mathbf{X} = \{x_0, x_1, \dots\}$ は観測系列であり、 x が観測されるたびに楽器のパートごとの状態 $\mathbf{Z}^{(k)} = \{z_0^{(k)}, z_1^{(k)}, \dots\}$ が遷移するモデルとなっている。これを楽譜追跡の問題に適用すると、 k は演奏パートの番号、 $z_n^{(k)}$ は楽譜上の音符、 x_n は観測された音高となる。観測にはMIDIの音高情報を用いた。

式で表すと以下ようになる。

$$p(\mathbf{X}, \mathbf{Z}^{(k)}) = p(z_0^{(k)}) \left[\prod_{n=1}^N p(z_n^{(k)} | z_{n-1}^{(k)}) \right] \prod_{n=1}^N p(x_n | z_n^{(k)}) \quad (1)$$

$$p(x_n | z_n^{(k)}) = \sum_{i=1}^I e_{ni}^{(k)} x_{ni} \quad (2)$$

ここで、 I はMIDIで表現できる音高の総数、 $e_{ni}^{(k)}$ は状態系列 k の n 番目の観測における音高 i の出力確率である。(1)式は一般的な隠れマルコフモデルの式であり、状態遷移確率の計算部と出力確率の計算部に分かれている。(2)式は隠れマルコフモデルにおける出力確率であり、音高情報を考慮した式で、本研究で提案するものである。 x_{ni} は n 番目の観測で音高 i が観測されると1、観測されていない音高は0とする。観測 x_{ni} と音高 i における出力確率 $e_{ni}^{(k)}$ の積をとり全音高について足し合わせることにより出力確率を表現している。状態系列の推定はViterbiアルゴリズムにより行った。

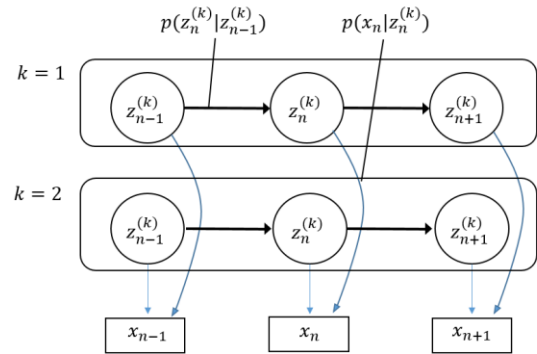


図1 Factorial HMM の例

3. 実験方法と結果

Factorial HMMを用いた楽譜追跡機能の確認実験を行った。1つ目に作成したMIDIデータを使用し、音高間違いが発生した際の動作確認、2つ目に人間が演奏したデータを用い動作確認を行った。音高間違いとは、演奏者が楽譜上とは異なった音高を演奏してしまうことである。

初めに、楽譜どおりに演奏したMIDIデータで楽譜追跡を動作させたところ、適切に追跡できることを確認した。次に、音高間違いが発生したことを想定し実験を行った。観測として、作成したMIDIデータから音高のみ抽出したものを使用した。音高間違いによる追跡結果の変化のみを確認するために遷移確率は固定し、出力確率を変化させて実験を行った。

楽譜と遷移確率について示す。使用する曲はモーツァルトのピアノ・ソナタ K. 545 の冒頭 2 小節を使用した。楽譜を図2に示す。状態系列は、右手のメロディ部の状態系列1と、左手の伴奏部の状態系列2を用意した。弾き飛ばし等に対応するために、遷移確率は正しい状態への遷移確率を0.8とし、0.2を他の全ての状態への遷移確率に振り分けた。音が発音されている途中で、もう一方の音に変化があった場合、自己遷移して状態を維持する。例えば状態系列1の最初の音符の場合には右手の2分音符が持続する間に左手の8部音符が3回演奏され、3回の自己遷移と1回の次の音符への遷移について、自己遷移確率を0.6、次の音符の状態への遷移確率を0.2と設定した。

†National Institute of Technology, Kushiro College

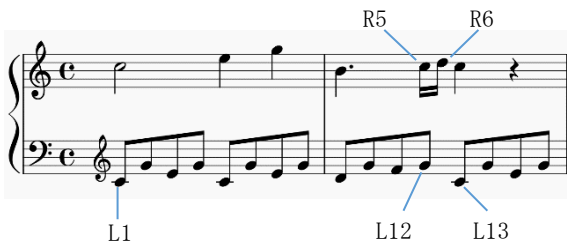


図2 実験に使用した楽譜

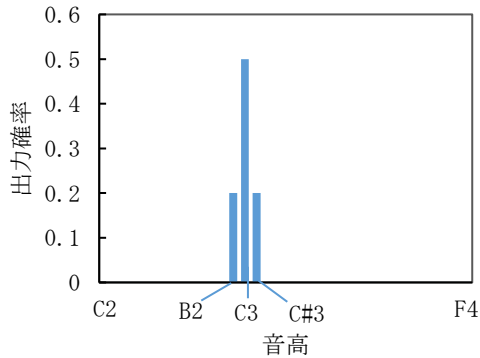


図3 音高間違いに対応した出力確率の分布

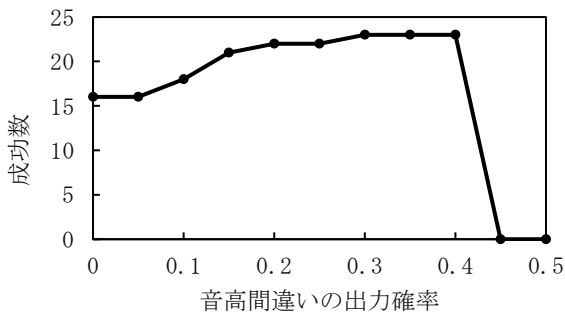


図4 楽譜追跡の成功数

演奏と出力確率について示す。音高を1か所だけ半音ずらしたデータを、全ての音符について作成した。正しい音高 i から半音ずれた音高 $i \pm 1$ の出力確率 $e_{n_{i \pm 1}}^{(k)}$ を「音高間違いの出力確率」と呼ぶことにする。音高間違いのデータは休符以外の音符数である23個用意した。正しい音高の出力確率と音高間違いの出力確率の合計が0.9となるよう出力確率を振り分けた。例えば、正しい音高の出力確率を0.6とすると、半音上と半音下の音高の出力確率はそれぞれ0.15と設定する。残りの0.1は他の全ての音高の出力確率に均等に振り分けた。楽譜上のある時点での音高がC3のときの出力確率を図3に示す。図3は正しい音高の出力確率を0.5とし、音高間違いの出力確率をそれぞれ0.2としたものである。音高間違いの出力確率 $e_{n_{i \pm 1}}^{(k)}$ を0から0.5まで0.05刻みで変化させたときの、楽譜追跡成功数の変化を図4に示す。成功数とは、音高間違いがない入力と同じように状態が遷移した回数である。図4から音高間違いの出力確率が高いほど音高間違いを追跡しやすくなることがわかる。音高間違いの出力確率が0.3付近で23個全ての状態遷移が成功した。

図3から音高間違いの確率が0.45付近で正しい出力確率が0になるので、楽譜追跡の成功数が0になることが確認できる。また、状態遷移系列を確認したところ、伴奏部は音高間違いの出力確率が低い場合でも、適切な結果が得られた。メロディ部は音高間違いの出力確率が0付近では成功しなかったが、確率が上がるにつれて成功数が増加した。これより、図2の楽譜における音高間違い時の出力確率を0.3付近に設定することで追跡機能が適切に動作するといえる。

次に、アマチュアのピアノ奏者の演奏により実験を行った。MIDIキーボードにより図2の楽譜を演奏したデータを入力とし、楽譜追跡の動作確認をした。演奏には音高間違いは含まれていない。出力確率は図3を用いた。MIDIデータの前処理について述べる。人間の演奏時に、複数音が同時に発音されていてもタイミングに小さなずれが生じ、観測数の増加に繋がっている。そこで、観測数が多くなることを防ぐために、前処理として演奏データのMIDIイベント間が30ms以下の場合には1つのイベントとして扱う処理[3]を行った。ここで、図2の楽譜の右手メロディ部の音符をからR1, R2, ..., R8と呼び、左手伴奏部の音符をL1, L2, ..., L16と呼ぶことにする。ここでは、休符も一状態であるから音符として扱う。楽譜追跡の結果として、1小節目において奏者が行った演奏どおりに追跡していることが確認できた。2小節目のメロディ部ではR5, R6の音符が飛ばされて追跡されていた。伴奏部はL13への遷移が適切に行われず、L12からL1に遷移しているのが確認できた。これはMIDI信号のnote onとnote offが密集した楽譜2小節目の一部が1つのイベントとして扱われたことで、適切な追跡がされなかったと考えられる。観測回数を確認したところ、作成したMIDIデータは17回に対して、演奏されたデータは15回であり、2回少なくなっていた。対応策として、前処理で扱った30msの時間を短くし観測回数を確保すること等があげられる。

4.まとめ

Factorial HMMを用いて多声音楽におけるパートごとの楽譜追跡のモデル化の手法の提案をした。実験では、作成したMIDIデータと人の演奏によってモデルの動作確認を行い、音高間違いの出力確率の設定をした。人が演奏したMIDIデータにおいて音の発音が密集する箇所では適切に楽譜追跡が動作しない場合があることがわかった。今後は弾き飛ばしに対応できる出力確率、状態遷移確率を設定していくことが必要である。

参考文献

- [1] 鈴木孝輔, 上田雄, 齋藤康之, 小野順貴, 嵯峨山茂樹, 「HMMを用いた音響演奏の楽譜追跡による弾き直しに追従可能な自動伴奏」, 情報処理学会研究報告, Vol. 2011-MUS-89, No. 29, pp. 1-6, 2011.
- [2] Z. Ghahramani, M. I. Jordan, "Factorial Hidden Markov Models", Machine Learning, Vol 29, Issue 2-3, pp. 245-273, 1997.
- [3] Diemo Schwarz, Nicola Orio, Norbert Schnell, "Robust Polyphonic Midi Score Following with Hidden Markov Models", International Computer Music Conference (ICMC), 2004.