

コード進行 HMM を用いた伴奏生成支援システムの構築*

小野 まなつ[†], 伊藤 克亘[†]

1 序論

音楽の世界において、曲はメロディと伴奏で構成されていることがほとんどであり、伴奏は和声法等を考慮した上でコード進行が組まれている。ただそのタイミングでメロディと伴奏の和音があてれば良いという訳ではなく、1つのコードに対してその前後への遷移の仕方も考慮する必要がある。実際に楽器を演奏できない人や音感、音楽経験が無い人であっても知っている曲のメロディを歌う事はできるが、伴奏をつけるとなるとそうはいかず専門知識が必要となってくる。伴奏にはメロディ単体では得られない響きを付けたり強弱・表情をはっきりさせる、メロディにおける休符を埋めるといった役割がある。また、1つの曲に対して原曲は1つであるが、その曲の伴奏のパターンや音色を工夫する事によって違う曲調にアレンジすることもできる。伴奏をつけるのと同様に知識無しにできることではないが、曲のアレンジが簡単にできれば演奏の幅を広げる事ができる。そこで本研究では、歌って入力されたメロディに対して既存の曲に基づいた伴奏の生成システムを考案する。

2 既存のコード提案システム

従来の伴奏生成システムとして Band In A Box、MySong[1] の2つが挙げられる。Band In A Boxには自動作曲や自動編曲の機能が備わっており、その中の1つとしてメロディからコードを提案するという機能がある。ここでのコードというのは、メロディに合った和音ではあるが、コード進行は考慮されておらず、曲を通して聞いた時に不自然な伴奏になってしまう事がある。対して MySong は既存の曲をデータベースとして学習しており、HMM[2] を利用して和音を選択しているためコード進行も考慮された伴奏となる。本研究では MySong と同様に HMM を利用した伴奏生成機能を作成し、その精度の向上を目標とする。

3 メロディに対するコードの選択方法

3.1 データベース

既存の曲の伴奏を学習内容とする。コードはメジャー、マイナーをはじめとしていくつも種類があり、さらにその種類×根音 12音分のコードが存在する。種類が多いほどコードのパターンの選択肢が増え、より原曲に使用されているコードを再現することが可能になる。しかしその分データベースの収集、処理速度、結果の精度に影響が出るため、使用するコードの数を軽減させる必要がある。データベースに含まれる各コードが使用されている回数を基に頻りに利用されているものを絞り、全ての音階についてのメジャー、マイナー、メジャーセブンス、ドミナントセブンス、マイナーセブンス、サスフォーの6種類を用いていく事にする。

*: Construction of accompaniment generation support system using chord progression HMM Manatsu Ono (Hosei Univ.) et al.

[†]法政大学 情報科学部

今回データベースは2種類作成する。1つはメロディとコードの関係を表したもの(データベース1)、もう1つはコードの遷移確率(データベース2)である。データベース1については、あるコードが演奏されている間のメロディに含まれている音の分布をまとめたものである。コードはメロディの音に合わせて決められているため、ここからあるメロディがあったときに演奏可能となるコードを選択していく。データベースとして利用する曲目に対して、コードが演奏されている区間に含まれるメロディの音階12個の回数を求める。

メロディの音がどのコードから出てきたと考えるのが尤もらしいか、というのが今回用いたい尤度である。図1のように、データベース1における割合がその音とコードとの尤度を表している為、1小節に含まれるメロディについて尤度を出したら対数を取り、足し合わせる。その際、尤度が0になった場合は対数尤度が極小さい値になるようにフロアリングしておく。

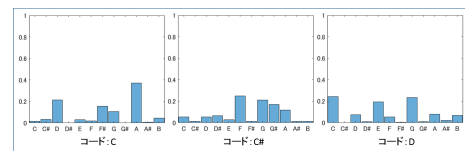


図1. コード毎のデータベースの分布(縦軸: そのコードに含まれるメロディの音の割合、横軸: メロディの音階)

データベース2では、使用されている全てのコードについて、あるコードからあるコードに遷移する確率を求めていく。この確率は、データベース1を基に絞った選択肢の中から、実際の曲の中で使用確率の高いコードの遷移先を選ぶために用いる。

3.2 メロディの音に合うコードの選択

使用可能なコードの候補をしばっていくために、メロディの音の分布を多項分布としてデータベースとの尤度を求める。入力されたメロディ1音ずつについて、その音がどのコードが出てきたと考えるのが尤もらしいか、というのが今回用いる尤度である。メロディの音がCだった場合、データベースの全てのコードのCのビンの割合が尤度となり、メロディ全てについてこの尤度を求める。尤度は値が小さくなりすぎないように対数をとって計算していく。

3.3 Ergodic HMM

メロディに含まれる音階を基に、演奏可能なコードを選択していく。ピッチ抽出の段階でコードの選択肢を絞った中からさらに1つ選び、伴奏としてコードを並べていく。ここで全状態へ遷移可能な ErgodicHMM[3] を利用する。

今回は図2のようなトポロジーの ErgodicHMM を利用する。HMMには初期状態の状態出力確率と各状態への遷移確率を与えておく。状態出力確率はそれぞれの状態の時にコードが出力される確率であり、初期

値は合計が1となるような乱数で与える。トポロジーとなる遷移確率は、同じ状態に帰ってくる確率、開始状態から直接終了状態へ行く確率、終了状態から他の状態へ行く確率は0としておく。この初期状態を基にデータベース2からコード遷移を学習し、学習後の状態出力確率と遷移確率を用いてコードの尤度を求めていく。

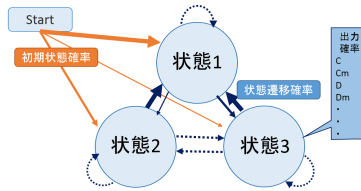


図 2. 3 状態の場合の Ergodic HMM

3.4 最終的な尤度の出力

1 小節毎にコードの尤度を出すために、各状態の出力確率にその小節のメロディに対する出力尤度を同じコード同士で足し合わせる。各状態で1番尤度が高いコードを選び、前の各状態からの遷移確率との和を取る。ここで図3のように状態数の2乗の数コードの候補が出てくるため、その中で尤度が最大のものを出力する。

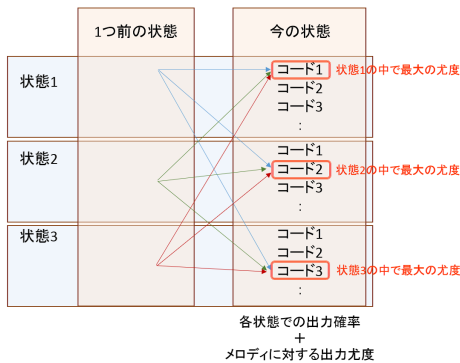


図 3. ErgodicHMM を用いたコードの尤度の求め方

3.5 結果出力

入力としてコードを付与したいメロディの音階、音価、最小音価、コードを何拍ずつで付与するかの情報を与えると、1 曲分のコード進行を出力する。結果はメロディと伴奏どちらも記載された楽譜を出力する。musicxml形式でデータを得られるようにして musescore で開くことで、聴覚的にユーザーが完成したものの確認でき、また、楽譜として表示することで伴奏生成後にユーザーが楽器演奏をする際にも利用できるようにするためである。出力には MuseScore を利用して楽譜の形式にし、システムによる伴奏生成後もユーザーが音の修正をできるようにする。コードの修正を行いたい場合は、コード遷移の学習を反映する前段階で候補として挙げられてい3つを提示し、修正出来るようにする。

4 評価、考察

メロディとコードの尤度の計算、HMM の状態出力確率と遷移確率を反映させた時に最も尤度が高くなる

コード進行の出力をするプログラムの動作確認を行った。プログラムを実行すると、出力結果は図4のようになった。

図 4. 1 曲分にコードを付与した例 (「It's a small world」)

緑の網掛けの部分が原曲と同じコードが付与された小節、赤の網掛けの部分が原曲とは違うコードが付与された小節であり、この曲の再現率は9/17となった。メロディとコードの尤度を確認してみると、原曲のコードが高い順位に含まれていた。学習後の HMM のを確認してみると、状態遷移確率は概ね正しく学習出来ているが、一部の入力のメロディのキーに対して本来は出てきにくいコードの確率が高くなってしまっていた。データベース2を作成する際に用いる曲のキーを考慮せず、色々なキーで出てくるコードが含まれている事が原因であると考えられる。これはデータベース2を全てCのコードのもの、又はCのキーに移調したもので作成することで改善が見込める。また、遷移確率についても偏りが見られた。この点に関して、同じコードであってもキーの違いによって扱われ方が変わってくるため、データベース2でのキーが問題であると考えられる。

5 おわりに

本稿では、あるメロディが与えられた時に伴奏として適切なコード進行を出力するシステムを作成した。今後は既存の曲と全く同じではなくても違和感のないコード進行が作成できているかを評価するために主観評価実験を行っていく予定である。課題としては、出力するコードの種類が増加があげられるが、ただ種類を増やすだけでは適切な選択肢に絞る中で精度が下がってしまう可能性がある、そのため、どの程度の頻度で出力させるのか、こういったパターンの場合に組み込んでいくのか等、更に条件を細かく指定していく事で、より原曲に近く、コードの幅が広がったコード進行が付与が可能になると考えられる。

参考文献

- [1] I.Simon et al., "MySong: Automatic Accompaniment Generation for Vocal Melodies", CHI '08 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2008, pp.725-734.
- [2] I.Simon et al., "Exposing Parameters of a Trained Dynamic Model for Interactive Music Creation"
- [3] 山本寛樹 他, "隠れマルコフモデルによる言語モデル自動獲得の検討", 情処学論, 1992, pp.227-228