旋律ベクトル表現を用いた事例ベースによる自動和声付け

松井　祥峰　池田　剛　乾伸雄　小谷　善行
東京農工大学　工学部　情報コミュニケーション工学科

１．はじめに

音楽情報処理の分野において、旋律の自動和声付けという研究は古くから行われている。その中で、コード進行を自動生成する研究については事例ベースによるものが多い。これらの研究では、事例を１/2 小節や１音符単位等の長さごとに区切って統計をとる手法が主流である[1]。

本研究では、コード変化する位置に着目し、事例を１つのコードの持続時間で区切って統計をとることによって、コード変化の自然なコード進行の生成を試みる。また、メロディーの種類は多様であり、それをカバーする事例を集めることは非常に困難である。そこで、メロディーをベクトル表現を用いて抽象化し、様々な入力に対してロバストに対応できるようにする。

２．コード進行の生成の手法

2.1　コード進行生成の確率的モデル

メロディーM $=\{m_0,m_1, m_2,\cdots,m_n\}$からコード進行 C$=\{c_0,c_1, c_2,\cdots,c_n\}$が生成される確率 P(C|M)は、Bayes の定理によって次のように表される。

$$P(C\mid M)=\frac{P(M\mid C)P(C)}{P(M)} \tag{1}$$

2.2　コード進行の確率的モデル

コード進行の確率的モデルには、コード進行のバイグラムを用いた。これにより、(1)式における P(C)は次のように近似できる。

$$P(C)\cong\prod_{i=1}^{n}P(c_i\mid c_{i-1}) \tag{2}$$

2.3　メロディーの確率的モデル

コード進行 C におけるメロディーM の存在確率 P(M|C)は、コード進行をユニグラムで近似した。これにより、メロディーの存在確率 P(M|C)は次のように近似できる。

$$P(M\mid C)\cong\prod_{i=1}^{n}P(m_i\mid c_i) \tag{3}$$

2.4　コード進行の生成

(1)式の分母はコード進行とは無関係であるため、P(C|M)は、(2)(3)式より次のように表される。

$$P(C\mid M)\cong\prod_{i=1}^{n}P(m_i\mid c_i)P(c_i\mid c_{i-1}) \tag{4}$$

ここで、(4)式の右辺の確率値を事例から抽出することによって、P(C|M)の最大値を求める。

2.5　メロディーのマッチング

$P(m_i|c_i)$は、メロディーm_i とマッチングした、事例にあるメロディーm´における P(m´|c_i)をもとに決定する。P(m´|c_i)は関数 Sr($0\le Sr\le 1$)を用いて次のように近似する。

$$P(m_i\mid c_i)\cong Sr(m_i,m')P(m'\mid c_i) \tag{5}$$

関数 Sr では、引数である二つのメロディーのマッチングを行っている。例えば、m_i と m´ が一致するとみなすことができれば Sr=1 であり、逆にそうでなければ Sr=0 となる。

３．ベクトル表現を用いたメロディーの抽象化

3.1　メロディーのベクトル表現

入力メロディーと対応する事例は、スパースネスを防ぐため、双方のメロディーをベクトルによって抽象化し、それを比較することでマッチングを行う。

メロディーを抽象化させるため、12 次元ベクトル V = ($v_0,v_1, v_2,\cdots,v_{11}$)を用意する。この 12 次元ベクトルを用いて、ドからシまで 12 音階の情報を格納する。また、各要素の大きさは、その音階の長さによって決定する。これにより、メロディーの情報は音高と長さのみとなり、音符同士の前後関係などは無視される。

実際の各次元におけるベクトルの大きさは、メロディーの構成音におけるその音高が、16 分音符単位で区切って何個分の長さかによって決定する。

例えば図２のようなメロディーは、次のようにベクトルで表現される。

Case-based Automatic Harmonization using Vector Expression of Melody.
Yoshitaka Matsui, Takeshi Ikeda, Nobuo Inui, Yoshiyuki Kotani.
Department of Computer, Information and Communication Sciences, Tokyo University of Agriculture and Technology.

(1,0,3,0,2,0,0,8,0,2,0,0)

図2．メロディー例

3.2　メロディーベクトルによるマッチング

メロディーをベクトルで表現することによって、メロディーの類似性を数値で容易に表現できるため、メロディーのマッチングは単に一致するかどうかだけでなく、それに近いものまでも抽出することが可能である。これにより、スパースネスを防ぐことができ、ロバストなコード進行の生成を実現している。

メロディーベクトルを用いることによって、(5)式における P(m´|c_i)は、V´をメロディーm´のベクトルとしたとき、次のように表される。

$$P(m'|c_i) = P(V'|c_i)P(m',V'|c_i) \qquad (6)$$

ここで P(V´| c_i)は、あらかじめ事例から求めることができる確率である。また、P(m´,V´|c_i)はコード c_i による条件付確率であるが、実際、ベクトル V´で表現されるメロディーの中に m´が存在する確率は、コードによる依存は少ないと考え、これを P(m´,V´)で近似する。この値は、コードによらないため、P(m´|c_i)は P(V´| c_i)で近似することになる。

また、(5)式における関数 Sr は、ベクトルの余弦によって計算できる。V_i をメロディーm_i のベクトルとしたとき、Sr を次のように定義した。

$$Sr(m_i, m') = \begin{cases} 1 & (\cos(V_i, V') \geq \varepsilon) \\ 0 & (\cos(V_i, V') < \varepsilon) \end{cases}$$

Sr によって、ベクトルの余弦が閾値εを超えた場合をマッチングの結果とする。また、閾値εに満たないものは、Sr=0 とし、候補から除外している。

ここで、今回は、εを 0.95 として実験を行った。

5．コード進行の生成実験及び結果

事例データ 30 曲を用いて、事例の長さを１/2 小節単位、１拍単位で区切った場合との比較実験を行った。また、評価の方法として、次のような項目について調べた。

1）コードネーム正解率

生成したコード進行を R={r_0,r_1, r_2,･･･,r_n}、正解のコード進行を A={a_0,a_1, a_2,･･･,a_n}としたとき、コードネーム正解率 Cp は次のようになる。

$$Cp = \frac{\sum_{i=0}^{n} S(r_i, a_i)}{n}$$

ただし、$S(a,b) = \begin{cases} 1 & (a = b) \\ 0 & (a \neq b) \end{cases}$ とする。

2）コード変化位置正解率

コード変化位置正解率 Pp は次のようになる。

$$Pp = \frac{\sum_{i=0}^{n} AND(T(r_i, r_{i-1}), T(a_i, a_{i-1}))}{\sum_{i=0}^{n} OR(T(r_i, r_{i-1}), T(a_i, a_{i-1}))}$$

ただし、$T(a,b) = \begin{cases} 1 & (a \neq b) \\ 0 & (a = b) \end{cases}$ であり、AND を論理積、OR を論理和とする。

オープンデータ７曲に対してコード進行の生成を行った結果、それぞれの正解率の平均値は表１のようになった。

表1．コード進行生成の結果

	コードネーム正解率	コード変化位置正解率
可変単位モデル	39.4%	47.4%
1/2 小節単位モデル	42.2%	33.8%
１拍単位モデル	28.2%	34.0%

6．まとめ

コードの持続時間で事例を区切った場合が、最もコード変化正解率が高かった。また、コードネーム正解率も、最も高かった 1/2 小節単位モデルに近い値となった。

7．おわりに

コード進行生成の精度は、全体的に低かったものの、可変単位モデルは、他のモデルと比べて優秀であった。また、メロディーをベクトルで抽象化することによってメロディーのマッチングがロバストに行えた。

参考文献

[1]志田裕樹：HMM を用いたベースラインからのコード進行の推定東京農工大学工学研究科電子情報工学専攻　2001 年度修士学位論文,2002