

F-25

クラシック曲のダイジェスト生成法の提案

Proposal of the digest generating method for pieces of classical music

堀野 義博
Yoshihiro HORINO黒木 進
Susumu KUROKI北上 始
Hajime KITAKAMI広島市立大学 情報科学部 知能情報システム工学科
{horino,kuroki,kitakami}@db.its.hiroshima-cu.ac.jp

1. はじめに

一般的な楽曲検索システムで、条件に合う楽曲を1曲に絞り込むことは、テキスト検索などの場合と同様に困難であり、最終的には各曲を聴取し選択する必要がある。そうした場合、利用者には時間的な負荷が大きくかかるという問題が生じる。そういった問題を解決するため、楽曲の繰り返し構造を利用して内部から特徴的な部分を抽出し、ダイジェストを作ることで楽曲選択に必要な聴取時間の短縮を試みた。

関連研究として、平田ら [1] が GTTM を用いた対話型の音楽要約システムを提案している。このシステムでは GTTM によって構造化された楽曲を元にして、対話的に音楽要約を行う。生成される曲は高品質であるが、そのためには数時間に及ぶ編集作業が必要とされ、楽曲データベースにおける膨大な数の楽曲には対応できない。本稿では、より簡易で高速なダイジェスト生成法を提案し、それについて検討・実験をした結果を報告する。

2. ダイジェストの生成

本研究では以下のような流れでクラシック曲のダイジェストを生成する。

1. クラシック曲の SMF (Source SMF) から旋律を抽出し、それを SMF として保存する (Melody SMF)。
2. 旋律の SMF (Melody SMF) から頻出区間を抽出し、上位 n 区間*の開始、終了時刻のリストを生成する (Timetable file)。
3. 区間時刻リスト (Timetable file) を参照し、元の SMF から頻出区間のみを取り出し、これらをダイジェストとして出力する (Digest SMF(s))。

以下では各段階での手法について詳しく説明する。

2.1 旋律抽出

本稿では、楽曲中の同時刻に発音される音のうち、最も音量が大きく、音高が高い音を1つだけ選択し、それによって得られる単音列を「旋律」と呼ぶことにする。

2.2 頻出区間の抽出

ここでは、得られた旋律のみの SMF を入力とし、楽曲内で互いに類似する小節を探し、各小節の出現頻度を調べる。小節の出現頻度を元に曲中の頻出区間を抽出する。

*抽出するフレーズ数 n は、現時点ではプログラム中で設定するようになっている。

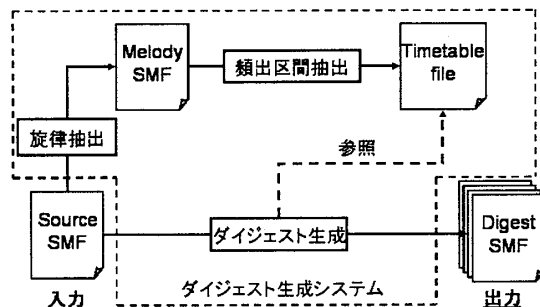


図 1: システム構成図

小節の類似判定 本稿では小節内の音列の時刻情報と音高変化情報を用いて類似判定を行っている。小節の類似判定は以下のような流れで行っている。

1. 発音時刻に基づいた音の対応づけ

注目している小節 A 内の i 番目の音 A_i の時刻を $t(A_i)$ 、比較する小節 B 内の j 番目の音 B_j の時刻を $t(B_j)$ とする。これらと比較し、その誤差 $\Delta t_{ij} = |t(A_i) - t(B_j)|$ が設定したしきい値 T_t 以下かどうかを調べる。 $\Delta t_{ij} < T_t$ の場合、音 A_i と B_j とは対応関係があると呼ぶことにする。これを以下の条件で進めていき、すべての音について時刻の対応関係を調べる。

$$\begin{cases} (1) & i++, j++ : \text{if } \Delta t_{ij} < T_t \\ (2) & i++ : \text{if } \Delta t_{ij} \geq T_t \text{ and } t(A_i) < t(B_j) \\ (3) & j++ : \text{if } \Delta t_{ij} \geq T_t \text{ and } t(B_j) < t(A_i) \end{cases}$$

対応音数 ((1) の場合) を C 、非対応音数 ((1) 以外の場合) を M とし、小節内の非対応音の割合 $M/(C+M)$ を計算し、これがしきい値 T_c 以下ならば次のステップへ進み、そうでない場合は類似していないと判定する。

2. 対応関係にある連続した2音の音高変化の照合

音 A_i と B_j が対応し、同様にその直後の音 A_{i+1} と B_{j+1} が対応しているとする。2音の音高 $p(A_i), p(A_{i+1})$ の変化量を $p(A_{i+1}) - p(A_i)$ とする。このとき、小節 A と B で音高変化の誤差 $\Delta p_{ij} = |(p(A_{i+1}) - p(A_i)) - (p(B_{j+1}) - p(B_j))|$ を計算する。対応のとれているすべての音についての Δp_{ij} を計算し、その平均 $\sum \Delta p_{ij} / (C - 1)$ が設定したしきい値 T_p 以下ならば小節 A と B は類似していると判定される。

頻出小節の検出 小節同士の類似性を調べ、各小節の類似小節の数が求められる。本稿ではこの類似小節の数を、その小節の出現頻度と呼ぶ。頻出区間の抽出の前の段階として、頻出小節の検出を行う。曲中で最も頻出する小節を探し、その小節を軸として頻出区間を形成していく。

頻出区間の抽出 頻出小節の前後も同じように類似関係があることが予想されるので、以下のように区間の伸張を行う。

1. 類似表¹の最頻出小節の位置を列方向に調べていき、1 (類似) が格納されているところを探索する。
2. そこから前後へ探索し、類似関係がなくなる、つまり0 (非類似) にぶつかるまで区間を伸張する。
3. 以上を列の最後まで繰り返す。区間の開始時刻、終了時刻は、区間が最も長くなるように選択する。

頻出区間の時刻情報の保存 前述の方法で頻出区間が一つ抽出される。この頻出区間の開始時刻と終了時刻をファイル (Timetable file) に保存する。

こうして、一つの頻出区間が抽出される。続いて、抽出された区間の頻度情報と類似情報を0にし、再度頻出小節の検出からの処理を抽出フレーズ数だけ繰り返す。最終的にn個の開始時刻と終了時刻の対がファイル (Timetable file) に出力される。

2.3 ダイジェスト SMF の生成

頻出区間の時刻情報を参照し、その区間内のイベント情報はそのまま出力し、それ以外のイベントは削除または一点に集約する。こうして入力 SMF (Source SMF) から各頻出区間のみの SMF (Digest SMF) が生成される。これらの SMF を聴取することで楽曲全体の特徴・印象をより短い時間で捉えることができる。

3. 評価実験

以下の5曲のクラシック曲の SMF を用意し、それぞれダイジェストを生成した結果について評価を行った。

表 1: 実験に用いた楽曲

	作曲者	曲名
1	Chopin	Waltz Opus 18 in E-flat major
2	Chopin	Waltz Opus 64 No.1 in D-flat major
3	Chopin	Etude Opus 10 No.3 in E major
4	Chopin	Etude Opus 10 No.12 in C minor
5	Satie	Je Te Veux

ここでは抽出されたフレーズが以下の条件を満たすかどうか調べた。

- 主テーマ (主題) 部分を抽出している。
- 主テーマ以外のテーマ (副テーマ) をある程度抽出している。

¹ 曲中の任意の2つの小節が類似しているかどうかを0と1で記録した2次元行列である。

結果の考察 実験した5曲において、曲の主テーマは抽出できることが確認できた。小節の類似判定時の各パラメータのしきい値 (T_t, T_c, T_p)[†] の値によって出現頻度が変わり、最終的な抽出結果に影響がでることが予測できる。ここでは音高変化パラメータのしきい値 T_p だけを変化させて、結果の違いを考察した[‡]。主テーマの抽出については、しきい値 T_p によらず、安定して抽出できた。それ以外の部分では、 T_p を1.0より小さい値に設定した場合は必要なフレーズが抽出されにくくなり、逆に T_p を1.0より大きくした場合は不要部分まで抽出されてしまった。これは、主テーマは曲中でもあまり変化 (変奏) はされず、それ以外の部分では曲に変化を与えるために変奏がほどこされているという傾向から、小節の類似関係の分布の形がテーマごとに異なり、それによって小節の出現頻度が変わってくるのが原因ではないかと考えられる。

圧縮率 提案する方法によって楽曲の演奏時間がどれくらい短縮されたかを表2に示す。

表 2: 抽出時間と圧縮率

	A	B	C	D	E
1	312	106	0.34	5	21
2	106	45	0.42	2	22
3	249	111	0.45	4	28
4	137	31	0.23	3	10
5	308	145	0.47	5	29

(A:全時間(秒) B:抽出時間(秒) C:圧縮率(B/A))
(D:フレーズ数 E:平均時間(B/D)(秒))

表2より、提案する手法によって楽曲の演奏時間を3~4割程度の時間に短縮できていることがわかる。

4. おわりに

本研究ではクラシック曲を対象とした楽曲のダイジェスト生成法を提案した。その結果として、楽曲の主テーマの抽出は期待どおりにでき、それ以外のテーマについても、しきい値を適切に設定すればうまく抽出できることがわかった。また、本手法によって楽曲を3~4割程度の時間に短縮できることがわかった。

現在の問題点として、小節のマッチングを行うときのしきい値 T_p の設定によって、抽出結果が変わってくるということが挙げられる。これは特に主テーマ以外の部分について、楽曲内の変奏の度合いによって類似小節の誤差の変動幅が変わってくるのが原因として考えられる。小節の類似度の定義を再検討し、誤差の変動に対応したしきい値の設定法について考えていくことが今後の課題である。

参考文献

- [1] 平田 圭二, 松田 周: 「パピプーーン: GTTM に基づく音楽要約システム」, 情処研報 MUS46-5, 2002.

[†] T_p の単位は半音を1とし、 T_t は SMF 内の4分音符長の分解能を1単位としている。

[‡] T_t は曲内での16分音符長、 T_c は0.2に固定した。