2J1-1in

MusiCube: 特徴量空間における対話型進化計算を用いた楽曲提示インタフェース

MusiCube: A Music Selection Interface featuring Interactive Evolutionary Computing in Feature Spaces

斉藤優理*1　伊藤貴之*1
Yuri Saito　Takayuki Itoh

*1お茶の水女子大学大学院　人間文化創成科学研究科
Graduate School of Humanities and Sciences, Ochanomizu University

We often want to select tunes based on our purposes or situations. For example, we may want background music for particular spaces. We think interactive evolutionary computing is a good solution to adequately recommend tunes based on users' preferences. This paper presents MusiCube, a visual interface for music selection. It applies interactive genetic algorithm in a multi-dimensional musical feature space. MusiCube displays a set of tunes as colored icons in a 2D cubic space, and provides a user interface to intuitively select suggested tunes. This paper presents a user experience that MusiCube adequately represented clouds of icons corresponding to sets of users' preferable tunes in the 2D cubic space.

1. はじめに

音楽配信サービスやポータブル音楽プレイヤーの普及，内蔵記憶装置の記憶容量の増大に伴い，個人の保有する楽曲数が膨大化している．現在，タイトルやアーティスト名などのメタデータに基づいた楽曲検索が主流である．しかし，ユーザが目的に応じて選曲する場合，例えばカフェやドライブの BGM の選曲のように，時間，場所，雰囲気などに合った楽曲を複数選ぶ場合，楽曲の特徴や印象などの情報に基づいた選曲が有効であり，かつ選曲にユーザの嗜好を反映させる必要があると考えられる．

本報告では，楽曲の雰囲気に基づき，ユーザの嗜好を反映させた楽曲提示インタフェース「MusiCube」(music + cube) を提案する．MusiCube は，楽曲ファイルから抽出した楽曲特徴量に基づき，対話型進化計算を用いてユーザの嗜好を学習し，その学習結果をＧＵＩで提示するインタフェースである．MusiCube を用いて選曲すれば，ユーザは早期に満足度の高い推薦結果を得るだけでなく，自身の嗜好がどのような楽曲特徴量に起因しやすいかを気づくことができると考えられる．

2. 関連研究

2.1 音楽推薦

”Islands of Music ”[1] は，心理モデルや自己組織化マップを用いて，楽曲群を地図上に配置するシステムである．ユーザは，楽曲の特徴や印象に基づいて平面上で楽曲を選択することができる．ただし，この手法は次元削減された楽曲特徴量を用いて平面上に楽曲を表示しているため，どの楽曲特徴量がユーザにとって重要であるかわかりにくいという問題点がある．

また，音楽推薦や類似曲検索システムのアルゴリズムとして，3 方向アスペクトモデル [2] やトリーベクトル量子化手法 (Tree-based vector quantization) [3] が適用されている．我々は，これらのアルゴリズムを MusiCube に適用すれば，より満足度の高い推薦楽曲を提示できる可能性があると考えている．

連絡先: 斉藤優理，お茶の水女子大学大学院　人間文化創成科学研究科，〒 112-8610 東京都文京区大塚 2-1-1，yuri@itolab.is.ocha.ac.jp

2.2 進化計算

対話型進化計算は，メロディー生成などにも適用されている．GenJim [4] は，ジャズのメロディー生成するシステムである．ユーザは，リズムとコード進行に基づいて生成したジャズメロディーを聴き，よいと思った瞬間に good ボタンを，不適切と感じたときは bad ボタンを押すことで評価する．これらの操作を繰り返すことで，ユーザは嗜好を反映させたメロディーを作曲することができる．

3. 提案内容

本章では，MusiCube の処理手順やユーザインタフェースについて説明する．

3.1 処理手順

MusiCube の処理手順は以下のとおりである．

1. 前準備として，楽曲データから楽曲特徴量を抽出する．
2. MusiCube は，1 で抽出した楽曲特徴量に基づいて，各楽曲をカラーアイコンで立方体領域内に表示する．
3. 楽曲を特定のカラーアイコンで提示する．
4. ユーザは，提示楽曲に対して評価を行う．
5. MusiCube は，ユーザの評価に基づいて嗜好を学習する．
6. 3〜5 を繰り返す．

3.2 ユーザインタフェース

MusiCube には，楽曲アイコン表示機能，楽曲提示機能，再生機能の 3 つの機能がある．

MusiCube の GUI 画面を図 1 に示す．画面の左側にカラーアイコン群を表示し，画面の右側に機能を切り替えるための 3 つのタブを表示する．

3.2.1 楽曲アイコン表示機能

ユーザは，任意の楽曲特徴量を変数として xy 軸に割り当てる．楽曲特徴量については，4.1 で詳しく説明する．MusiCube は，ユーザによって設定された xy 座標平面上に各楽曲をアイコンで表示する (図 2(左) 参照)．ユーザは，xy 軸に割り当て

図 1: MusiCube の Window 画面：Window(左) 各楽曲を立方体領域内（特徴量空間内）にカラーアイコンで表示する．Window(右) 機能を切り替えるための 3 つのタブがある．

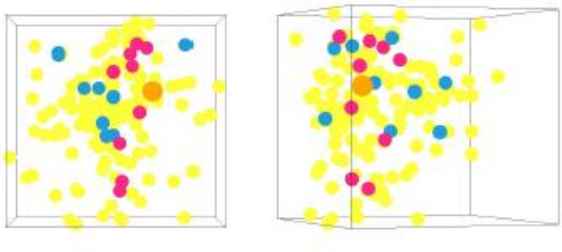

図 2: (左) 楽曲アイコン表示，(右)xy 軸に割り当てる楽曲特徴量を変化

る楽曲特徴量を変化させることで，さまざまな楽曲特徴量の組み合わせでアイコン群の分布を見ることができる (図 2(右) 参照)．

3.2.2 アイコンの色の割り当て

MusiCube では，各楽曲をカラーアイコンで表示する．アイコンの色は，以下のように割り当てる．

- 赤 ： ユーザが目的に合っていると評価した楽曲．
- 青 ： ユーザが目的に合っていないと評価した楽曲．
- オレンジ ： MusiCube によって提示された楽曲．
- 黄色 ： ユーザがまだ聴いていない楽曲，または評価していない楽曲．

3.2.3 楽曲提示機能

ユーザは，MusiCube が出力として提示する楽曲に対して，目的に合っているか否かについての主観的評価結果を 2 値で入力する．MusiCube は，ユーザの楽曲への評価に基づき，対話型進化計算を行い，次の出力として，よりユーザの目的に合っていると推定される楽曲を提示する．これらの操作を繰り返すことで，MusiCube は，ユーザの目的に合った楽曲を効率よく提示できるようになると考えられる．

評価が集まったら，ユーザは対話型進化計算を一時停止し，任意の楽曲特徴量を xy 軸に割り当て，評価済みの楽曲間の相関を見ることで，素早く目的に合った新しい楽曲にたどりつけることが予測される．

また，MusiCube には評価に基づいてユーザにとって最適な楽曲特徴量を選択する機能がある．最適な楽曲特徴量の選択については 4.3 で詳しく説明する．

3.2.4 再生機能

MusiCube は各楽曲に対する評価に基づき，自動的にプレイリストを作成することができる．ユーザは，立方体領域内に表示されたアイコン群上で，マウスクリックした位置の周辺にある赤色アイコン（目的に合っている楽曲）や黄色アイコン（まだ聴いていない楽曲，または評価していない楽曲）に対応する楽曲群を自動的にプレイリストに追加することができる．これにより，ユーザは雰囲気に合った楽曲を簡単に探すことができ，未知の楽曲との出会いを楽しむことができる．

4. MusiCube の実装

MusiCube を実現するためには，各楽曲から特徴量を抽出し，各インタフェース機能を実装する必要がある．

4.1 楽曲特徴量の抽出

我々は，MIRtoolbox [*1]を用いて，各楽曲から 10 個の楽曲特徴量を検出している（表 1 参照）．本手法では，これらの楽曲特徴量を同等に扱うため，楽曲特徴量 f を正規化した f' を用いる．ここで，$f' = (f - f_{min})/(f_{max} - f_{min})$ であり，f_{max} と f_{min} は楽曲特徴量の最大値と最小値である．楽曲には，次第に曲の印象が変わるものや，1 つの楽曲に複数の印象を合わせ持った曲があり，楽曲全体の特徴量を一つに定めるのは困難である．そこで，一定区間ごとに特徴量を検出し，次第に楽曲が変化する場合は各区間の特徴量の平均値をその楽曲全体の特徴量とし，特定の時刻から明確に印象が変わる楽曲に対しては，印象が変化する前と後に分けて複数の特徴量を 1 つの楽曲の特徴量とする．現段階では，楽曲の中間部のある 10 秒間から得た特徴量を楽曲全体の特徴量として暫定的に定めている．

表 1: 楽曲特徴．

特徴量	説明
RMS energy	音量
Low energy	弱音の割合
Tempo	テンポ
Zero crossing	波形が 0 の値を取る回数
Roll off	85%を占める低音域の値
Brightness	1500Hz 以上の割合
Roughness	不協和音の多さを示す値
Spectral irregularity	音質の変化の大きさ
Inharmonicity	根音に従っていない音の量
Mode	長和音と短和音の音量の差

4.2 対話型進化計算を用いた楽曲提示

MusiCube は，正規化した特徴量空間において，対話型進化計算の一つである対話型遺伝的アルゴリズム (interactive Genetic Algorithm: iGA) を用いて，ユーザの嗜好を反映し

*1 MIRtoolbox:
http://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox

た楽曲提示を行う．ただし，遺伝的操作を行う際，変数が多くなると，学習問題は難しくなる傾向があるため，本手法ではあらかじめ主成分分析によって，次元を削減している．

1. 初期個体生成と提示
   MusiCube は全楽曲の中から，あらかじめ定められた数の初期個体 (提示楽曲) を選択する．
2. 評価と選択
   ユーザの心理的な負担を考慮し，ユーザは提示楽曲に対して 2 値（目的に合っているか否か）で評価を行う．選択では，ユーザが「目的に合っている」と評価した個体（楽曲）を親個体とする．
3. 交叉
   親 2 個体間の範囲の中間値を子 2 個体の値とする．
4. 突然変異
   突然変異率を 0.1 とし，値をランダムに変化させる．
5. マッチング
   4 で得た子個体と個体（楽曲）のユークリッド距離 d を計算し，d が最小である個体（楽曲）を次世代に残す個体（楽曲）とする．

$$d = \sqrt{\sum_{i=0}^{n} (f_i - p_i)^2} \tag{1}$$

ここで，f は遺伝的操作で得られた楽曲特徴量，p は楽曲データの楽曲特徴量とする．

4.3 xy 軸に割り当てる最適な楽曲特徴量の選択

MusiCube では，目的に合っている楽曲を表す赤色のアイコンが集まるような 2 つの楽曲特徴量を提示するため，エントロピーを用いて，楽曲群の密度を計算している．まず，特定の 2 つの楽曲特徴量を xy 軸に割り当て，その表示領域内を N*N で分割する．次に，各分割領域内の赤色のアイコンと赤色以外のアイコンを数え，以下の式よりエントロピーの合計を算出する．

$$\begin{aligned} E_{sum} &= \sum^{N^2} (p_{ri} \log p_{ri} + p_{qi} \log p_{qi}) \\ p_{ri} &= r_i/(r_i + q_i) \\ p_{qi} &= q_i/(r_i + q_i) \end{aligned} \tag{2}$$

ここで，i-th 番目の分割領域における p_{ri} と p_{qi} は赤色のアイコンと赤色以外のアイコンの確率とする．本手法では，すべての楽曲特徴量の組み合わせにおいてエントロピーを算出する．最も小さい値となった楽曲特徴量の組み合わせをユーザにとって最適な楽曲特徴量とし，図 1 の Tab(b) においてそれぞれに対応する楽曲特徴量を特定のカラーフォントで提示する．

5. 実験

5.1 実験方法

我々は，Java JDK 1.6.0 を用いて MusiCube を実装し，Lenovo ThinkPad T510(CPU 2.4GB，RAM 2.0GB) および Windows7 の上で実装した．また，本実験において RWC [*2] 音楽データベースの 143 曲 11 ジャンル (ポップス，ロック，ダンス，ジャズ，ラテン，クラシック，行進曲，ワールド，声楽，邦楽，アカペラ) を用いて，13 人の被験者に対し，以下の実験を行った．

Step1 ： 被験者は MusiCube によって提示された楽曲を聴く．
Step2 ： 提示楽曲に対して，カフェの BGM として合うか否かを 2 値で評価する．
Step1〜2 を繰り返し，その後アンケートに答えてもらった．

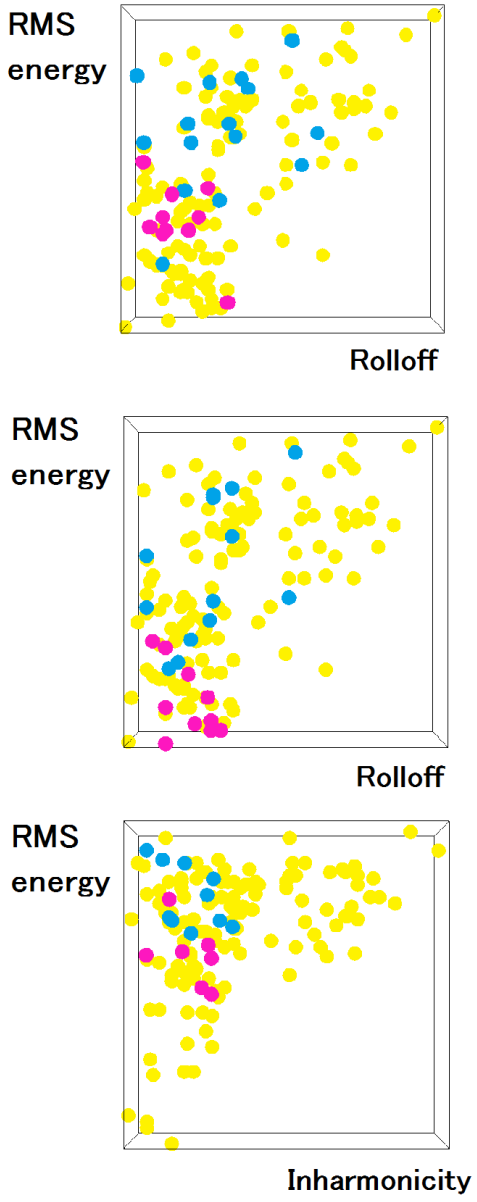

図 3: 被験者 3 人の実験結果

5.2 実験結果

図 3 に被験者 3 人の結果を示す．図 3(上) と図 3(中央) は，xy 軸には RMS energy と Rolloff が割り当てられ，アイコンの配置が類似していることから，2 人の嗜好が似ていると考えられる．一方，図 3(下) は，図 3(上) や図 3(中央) と異なり，xy 軸には RMS energy と Inharmonicity が割り当てられてい

*2 RWC Music Database: http://staff.aist.go.jp/m.goto/RWC-MDB/

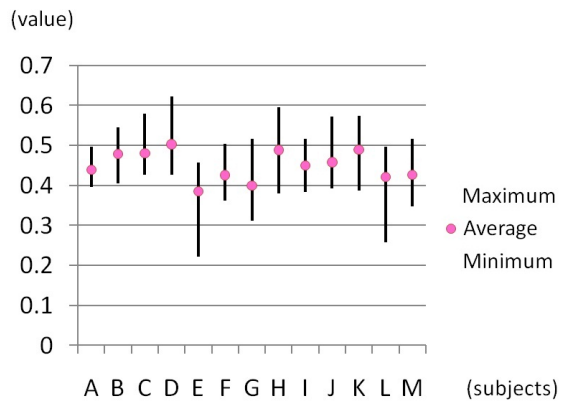

図 4: 被験者 13 人 (A-M) の結果における E_{sum} の最大値，平均値，最小値.

る．このことから，図 3(下) の被験者は，図 3(上) や図 3(中央) の被験者らと異なる嗜好を持っていることがわかる．つまり，カフェの BGM を選ぶ場合，個人の好みを考慮する必要があり，MusiCube のようなユーザの嗜好を考慮した選曲システムが有効であると考えられる．

また，我々は各被験者の結果において，xy 軸に割り当てる楽曲特徴量のすべての組み合わせでエントロピーを計算した．図 4 は，13 人の被験者のエントロピーの最小値，平均値，最大値を示している．すべての被験者において，ランダムに楽曲特徴量を選択するより，最適な 2 つの楽曲特徴量を xy 軸に割り当てた方が，赤色のアイコンは集まりやすくなることがわかった．

5.3 フィードバックと考察

我々は，被験者に MusiCube で音楽を聴いてもらい，以下の質問をした．

1. Q1：プレイリストにはカフェの BGM としてふさわしい楽曲は何曲ありましたか？
2. Q2：MusiCube は，音楽推薦の GUI として効果的でしたか？(5 段階評価)
3. Q3：どのような効果がありましたか？
4. Q4：MusiCube の改善点を教えて下さい.

Q1 では，各被験者に対し，プレイリストの全曲のうち，目的に合うと評価した楽曲を数え，割合を計算したところ，最大値 1.0，平均値 0.7，最小値 0.27 となった．我々は，0.5 以下の値となった 1 人の被験者を除き，全体として良い結果となったと考えている．

Q2 では，全被験者の平均評価値は，4.15 となった．この結果から，ほとんどの被験者が MusiCube を使うことにより，雰囲気に合った楽曲を選びやすくなったことを確認できた．

また，Q3 で得られた回答を以下に示す.

1. アイコンが密集していたため，聴きたい楽曲を簡単に探すことができる.
2. MusiCube のような視覚的な推薦システムは，自ら楽曲を探さないリスナーにとって，とても便利である.
3. 提示された楽曲が自分の好みと非常に合っていた.

これらのコメントから MusiCube の有効性を確認することができた.

Q4 で得られた回答を以下に示す.

1. 選曲する時，メタデータ (タイトルやジャンルなど) を利用すると，さらに便利になるかもしれない.
2. アイコンが重なり，各楽曲を見分けづらい.
3. 可視化結果によって，楽曲の感じ方が変わり，評価に影響される可能性がある.

我々は，これらのコメントを参考に MusiCube をより使いやすい音楽提示インタフェースとして改良していきたいと考えている.

6. まとめと今後の課題

本報告では，楽曲の雰囲気に基づきユーザの嗜好を反映させた楽曲提示インタフェース「MusiCube」を提案した．MusiCube は楽曲特徴量に基づき，対話型進化計算を用いてユーザの嗜好を学習し，その学習結果をＧＵＩで提示するインタフェースであり，ユーザは早期に満足度の高い推薦結果を得るだけでなく，自身の嗜好がどのような楽曲特徴量に起因しやすいかを気付くことができる.

以下に本研究の今後の課題を示す.

推薦精度の向上

我々は，4.1 節で論じた楽曲特徴量に加え，ジャンルなどのメタ情報や他者の嗜好情報などを利用し，推薦精度の向上を目指したいと考えている.

アイコンの配置

MusiCube は多次元からなる楽曲特徴量を次元削減することなく，任意の 2 つの楽曲特徴量を xy 軸に割り当て，ユーザにとって重要な楽曲特徴量を気付かせるインタフェースとなっている．しかし，楽曲特徴量を次元削減しアイコン群を表示した方が有用な場合がある．そこで我々は，目的に合っている楽曲を表す赤色のアイコンが集まるような結果を表示するため，線形判別分析を適用したいと考えている.

参考文献

[1] E. Pampalk, Islands of Music: Analysis, Organization, and Visualization of Music Archives. Master's thesis, Vienna University of Technology, 2001.

[2] 吉井和佳, 後藤真孝, 奥乃博, ハイブリッド型音楽推薦システム, CrestMuse Symposium 2008, 27-28, 2008.

[3] K. Hoashi, K. Matsumoto, N. Inoue, Personalization of user profiles for content-based music retrieval based on relevance feedback, Proc. of 11th ACM Int'l Conference on Multimedia, 110-119, 2003.

[4] J. A. Biles, GenJam: A Genetic Algorithm For Generating Jazz Solos, Int'l Computer Music Conf.(ICMC'94), 131-137, 1994