

# 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

理学専攻 情報科学コース 齊藤 優理

## 1 概要

デジタルオーディオプレイヤーや音楽配信サービスの普及により、ユーザは聴きたい楽曲を大量のコンテンツの中から探すことができるようになった一方で、聴きたい楽曲を探しあてることが難しくなってきた。

本論文では、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、楽曲データ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ことができ、さらに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把握することができる。

## 2 提案内容

### 2.1 処理手順

MusiCube の処理手順は以下のとおりである。

1. 楽曲データから特徴量を抽出する。
2. 特徴量空間に楽曲をアイコンで表示する。
3. 楽曲を特定のアイコンで提示する。
4. ユーザに提示楽曲を聴かせて評価させる。
5. 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。

### 2.2 主な機能

MusiCube には、楽曲アイコン表示、楽曲提示、再生の3つの機能がある。

#### 2.2.1 楽曲アイコン表示機能

ユーザは、各楽曲が有する多次元からなる楽曲特徴量(3.1 参照)の中から、任意の2次元(楽曲特徴量)を選択し、それらの楽曲特徴量を図1(左)に示すような ScatterPlot の  $xy$  軸に割り当てる。MusiCube は、ユーザによって設定された  $xy$  座標平面上に楽曲特徴量の値を座標値として各楽曲をアイコンで表示する(図1(左) 参照)。また、 $xy$

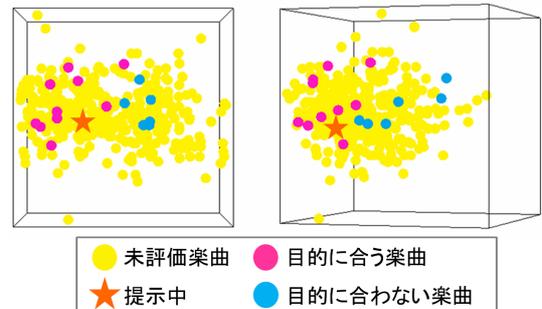


図 1: (左) 楽曲アイコンを表示, (右)  $xy$  軸に割り当てる楽曲特徴量を変化させる回転操作のアニメーション表示

軸に割り当てる楽曲特徴量を変化させる回転操作により、ユーザは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1(右) 参照)。

#### 2.2.2 楽曲提示機能

MusiCube では、対話型進化計算を用いてユーザの嗜好を学習し、最適な楽曲を提示する。ユーザはオレンジ色のアイコンで提示された楽曲を聴き、目的に合っているか否かで評価を行う。目的に合っていると評価された楽曲は赤色のアイコンに変わり、目的に合っていないと評価された楽曲は青色のアイコンに変わる。このように、“MusiCube による楽曲提示”と“ユーザによる楽曲評価”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。

楽曲への評価が集まったら、ユーザは対話型進化計算を一時停止し、立方体領域内における評価済みの楽曲間の相関を見ることができる。これにより、ユーザは自身の嗜好の傾向を把握することができる。また、MusiCube には目的に合っていると評価された楽曲群が立方体領域内で最も局所集中して表示されるような楽曲特徴量を提示する機能がある。

#### 2.2.3 再生機能

MusiCube では、各楽曲に対する評価に基づいてプレイリストを作成する。ユーザは、立方体領域内において、マウスクリックした位置の周辺に配置された楽曲群をプレイリストに追加することができる。これにより、大量の楽曲データの中から自身の気分やその場の雰囲気に応じて直感的に選曲することが可能となる。

### 3 MusiCubeの実装

#### 3.1 楽曲特徴量の抽出

本手法では、MIRtoolbox [1] を用いて、各楽曲から以下の楽曲特徴量を抽出している（表 1 参照）。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。

表 1: 楽曲特徴

| 特徴量                   | 説明                  |
|-----------------------|---------------------|
| RMS energy            | 音量の平均               |
| 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       |
| 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                  | major と minor の音量の差 |

#### 3.2 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) [2] を用いて、ユーザの嗜好を反映させた楽曲提示を行う。本手法では、遺伝的操作を行う際、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

ユーザに提示楽曲を 2 値 (目的に合っているか否か) で評価させる。選択ではユーザによって「目的に合っている」と評価された個体 (楽曲) を親個体とする。

##### 3. 交叉

親 2 個体間の中間値を子 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

以下の式によって、4 で得た子個体と個体 (楽曲) のユークリッド距離  $d$  を算出し、 $d$  が最小となる個体 (楽曲) を次世代に残す個体 (楽曲) とする。

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

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、 $n$  は染色体の長さ (遺伝子長) とする。

### 4 結果

MusiCube を使って被験者に「カフェの BGM に合う楽曲」を選曲してもらった。図 2 に被験者 3 人の実験結果を示す。なお、各図は赤色のアイコン (目的に合っている楽曲) 群の密度がより高くなる楽曲特徴量を  $xy$  軸に割り当てた表示結果である。図 2(左) と図 2(中央) は、 $xy$  軸に Rolloff と RMS energy が割り当てられ、アイコンの配置が類似している。このことから、図 2(左) と図 2(中央) の被験者らは類似した嗜好を持っていることが考えられる。一方、図 2(右) は  $xy$  軸に Inharmonicity と RMS energy が割り当てられている。このことから、図 2(右) の被験者は図 2(左) や図 2(中央) の被験者らと異なる嗜好を持っていることが考えられる。これらの結果より、気分や雰囲気に対応しい楽曲を選ぶ場合、ユーザの嗜好を考慮する必要があり、ユーザの嗜好がどのような特徴に起因しやすいかを気づかせる MusiCube のような選曲システムが有用であることがわかった。

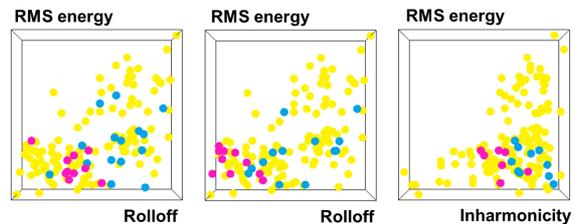


図 2: 被験者 3 人の実験結果

### 5 まとめ

本論文では、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る。評価実験結果では、MusiCube を用いることでユーザは早期に満足度の高い推薦結果を得ることができ、さらに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把握できることを示した。

### 謝辞

本論文の実験には、RWC 研究用音楽データベースと文京区立図書館に所蔵されている CD に収録された楽曲を使用させて頂きました。

### 参考文献

- [1] O. Lartillot. MIRtoolbox. <http://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox>.
- [2] H. Takagi. Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation. In *Proceedings of the IEEE*, 89(9), 1275–1296, 2001.