



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*

斉藤 優理¹⁾, 伊藤 貴之²⁾

MusiCube : A Visual Music Recommendation System featuring Interactive Evolutionary Computing

Yuri SAITO, Takayuki ITOH

ABSTRACT

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 real-valued genetic algorithm in a multidimensional 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.

Keywords : Music recommendation, Interactive evolutionary computing, Multidimensional feature space

1. はじめに

モバイル環境や音楽配信サービスの普及, および記憶媒体の低価格化により, 誰もがいつでもどこでも世界中の楽曲を聴くことができるようになり, 個人のデジタル機器に数千曲もの楽曲を保存できるようになった. その結果として, 聴きたい楽曲を探しにくい場合もしばしばある. 一般的に楽曲の検索には, タイトルやアーティスト名などのメタデータが手がかりとなる. それに対して我々は, メタデータに頼らずにユーザの嗜好に合わせて新しい楽曲を簡単に探すことができる楽曲提示インタフェースを開発することで, 音楽の楽しみ方を無限に広げていきたいと考えている.

事前調査として我々は, 236 人の学生に対して「気分に合わせて音楽を聴くことはあるか」というアンケートを実施してみたところ, 219 人が Yes と答えた. その中でも, 「気持ちを落ち着かせてリラックスしたい時」にはクラシック調の曲, 「就寝時」には静かで穏やかな曲, 「カフェにいる時」にはジャジーな曲, 「部屋の掃除をする時」にはアップテンポで明るい曲などのように, その時の気分合った楽曲を選んだり, YouTube などの類似曲検索を利用して音楽を聴いている人が多かった. このことから, その時の気分に応じた

楽曲を選択できる技術は重要であると考えられる.

大量の楽曲の中から, ユーザに適切な楽曲を推薦するさまざまな手法が近年提案されている. ユーザに適切な楽曲を推薦する主な手法に, 協調フィルタリングと内容 (楽曲の音響的な特徴など) に基づくフィルタリングがあげられる. 協調フィルタリングは, ユーザの嗜好情報を収集し, 嗜好の似た他のユーザが好む楽曲を推薦する. この手法は, ユーザにとって最適な楽曲を比較的精度よく推薦できるが, 主な推薦対象曲は多くのユーザに聴かれている楽曲となる傾向にあり, 推薦結果に嗜好の個人差や変化を反映することが難しい. 一方, 内容に基づくフィルタリングは, アーティストやジャンルに捉われず, ユーザの嗜好に合った楽曲と類似した曲調を持つ楽曲を推薦する. この手法は, ユーザのコンテンツの感じ方の違いを考慮して楽曲を推薦することができる. しかし, システムによる推薦結果にユーザが納得できない場合が多く, 議論の余地が残されている.

そこで本論文では, 内容に基づく対話的な楽曲フィルタリングの一手法として, アーティストやジャンルに捉われず, 楽曲の特徴に関するユーザの嗜好を反映させて選曲をするための楽曲提示インタフェース「MusiCube」を提案する. MusiCube では例えば, カフェやドライブ, ランニングの BGM (Back ground Music) を選ぶ時のように, ユーザがその時の場面設定に合わせて楽曲を選択する際に, アーティストやジャンルに捉われず, ユーザの嗜好に合っている楽曲をプレイリストにしたり, 知っている曲やお気に入りの曲の他にメロディーやタイトルさえも知らない楽曲を推薦する. これによりユーザは, 思いがけない楽曲を発見し, 音楽をより広く楽しむこと

* 原稿受付 ****年**月**日

1) 非会員 お茶の水女子大学大学院人間文化創成科学研究科
2) 正会員 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, E-mail: itot@is.ocha.ac.jp)

ができると期待される。また、音楽は聴く人によって感じ方が異なるため、他者の推薦情報は必ずしも信頼できるとは限らない。そこで MusiCube は、ユーザと計算機の自然なインタラクションによってユーザの嗜好を学習し、満足度の高い推薦結果を提示すると同時に、推薦結果の可視化をもってユーザに自身の嗜好の傾向を気づかせるユーザインタフェースとしても機能する。

MusiCube では前処理として、楽曲データ (wave 形式) から多次元の楽曲特徴量を抽出する。ユーザは、抽出した楽曲特徴量の中から任意の2次元の楽曲特徴量を選択し、それらの楽曲特徴量を Fig.1 に示す描画領域の xy 軸に割り当てる。続いて MusiCube は xy 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。アイコンには「目的に合う楽曲」、「目的に合わない楽曲」、「提示中の楽曲」、「未評価の楽曲」の4種類が用いられる。ここで、座標の2軸に割り当てる次元 (楽曲特徴量) を切り替えると、アニメーション表示によってアイコンを再配置することができる。

MusiCube ではユーザの嗜好を学習する手段として、人間の感性を交えた最適化手法の1つである対話型進化計算を適用する。MusiCube はユーザに提示楽曲を聴かせ、「目的に合う」「目的に合わない」の2択で評価させる。この「楽曲の提示」と「ユーザによる楽曲への評価」を繰り返すことで、ユーザの嗜好を学習する。以上により、MusiCube は大量の楽曲の中から「目的に合う」と予測される楽曲を提示すると同時に、学習過程と評価済みの楽曲の相関を可視化できる。そしてその過程で MusiCube は、「目的に合っている」と評価された楽曲群が画面上で最も局所集中して表示される次元 (楽曲特徴量) を提示する。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次元の楽曲特徴量を可視化することで、ユーザにとって重要な楽曲特徴量を気づかせることが可能になる。

対話型進化計算を用いた楽曲推薦の関連研究と比べて MusiCube には、特徴量空間における学習結果の可視化を積極的に導入することで、ユーザ自身の選曲傾向を気づかせ、選曲にある程度の一貫性を維持できるようなガイドラインを与えている点に特徴がある。

我々は、RWC 研究用音楽データベース[1]の楽曲データなどを適用事例として、MusiCube の効果を検証した。本論文ではその可視化結果から得られた知見を紹介するとともに、被験者実験結果から検証された MusiCube の有用性を示す。

2. 関連研究

本章では、音楽推薦と多次元データ可視化に関する従来手法、対話型進化計算を応用した事例について述べる。

2.1 音楽推薦

音楽を対象とした推薦システムには、プレイリストの自動生成や状況に応じた音楽推薦などがある。

音楽推薦の手法として、あるユーザに対し好みの似た他のユーザが好む楽曲を推薦する「協調フィルタリング (Collaborative Filtering)」や、あるユーザに対しそのユーザが好む楽曲と音楽内容が類似した楽曲を推薦する「内容に基づくフィルタリング (Content-based Filtering)」、これらを組み合わせた「ハイブリッド型

フィルタリング」が挙げられる[2]。

協調フィルタリングは、楽曲のみならず本や映画などさまざまな情報推薦サービスで採用されている。この手法では、ユーザの嗜好を予測するために必要なデータを集めることができれば、ユーザの好みに合うコンテンツを精度よく推薦することができる。しかし、数百万を超えるコンテンツを対象とする場合、各コンテンツの組み合わせパターンが非常に多くなってしまいうため、嗜好の予測に十分なデータを集めることが難しい。そのため、推薦の対象になる楽曲は主に多くのユーザに聴かれているごく一部の楽曲に限られることもよくある。また、楽しい曲やリラックスした曲というコンテンツの感じ方には個人差や変化があるにも関わらず、そのような嗜好の差や変化が推薦に反映されにくいなどの問題が指摘されている[3]。

そこで、このような問題を解決するために、内容に基づくフィルタリングが活発に研究されている。その多くは、ユーザが好む楽曲と音楽内容が類似している楽曲を推薦する。利用する音楽内容は、音楽音響信号に基づくもの[4]と楽曲の音楽要素に対するアノテーションなどのメタデータに基づくもの[5]に大別できる。ここで内容に基づくフィルタリングでは、ユーザの嗜好情報をモデル化する必要がある。嗜好をモデル化するアプローチには、特徴量空間においてユーザの嗜好をベクトルで表現するベクトル空間モデル[4]や、ユーザの嗜好を人間の主観を用いてモデル化するアプローチがある[6][7]。MusiCube では、音楽音響信号に基づいて、対話型進化計算[8]の一つである対話型遺伝的アルゴリズムを用いて、ユーザとのインタラクションによってユーザの嗜好情報を学習し、満足度の高い推薦結果を提示すると同時に、その学習結果を可視化することで、ユーザ自身に嗜好の傾向を気づかせるシステムになっている。

2.2 音楽の可視化

大量の楽曲を短時間で俯瞰し聴きたい楽曲を素早く見つけるために、楽曲の類似度や分類結果に基づいて画面上に楽曲群を配置し、視覚的に楽曲を推薦・提示する手法が活発に研究されている。楽曲群を楽曲の特徴に基づいて二次元または三次元座標上に配置するために、自己組織化マップ (Self-Organizing Map : SOM) [9][10][11]、主成分分析 (Principal Component Analysis: PCA) [12]、多次元尺度構成法 (Multi-Dimensional Scaling : MDS) [13]などの次元削減手法がよく用いられる。

ただし、これらの手法は次元削減された楽曲特徴量を用いて楽曲を座標上に表示しているため、ユーザにとって重要な楽曲特徴量がわかりにくいという問題点がある。本論文が提案する MusiCube では、多次元からなる楽曲特徴量を次元削減することなく、ユーザが多次元からなる楽曲特徴量の中から任意の2次元の楽曲特徴量を対話的に選ぶことで、ユーザにとって重要な楽曲特徴量を気づかせることが可能となる。

2.3 多次元データの可視化

多次元データの可視化手法は、変数間の相関性を分析するために有用な技術であり、顧客データや測定情報、シミュレーション結果の分析など幅広い分野で活用されている。

多次元データを可視化する代表的な手法として、Parallel Coordinates [14] や Scatterplot Matrix がある。Parallel Coordinates は、多くの変数にわたる統計的な傾向を把握しやすいが、表示結果が煩雑になりやすいという問題点がある。Scatterplot Matrix は、任意の2変数の相関性を把握しやすいが、次元の高いデータを適用する場合、表示領域が大きくなってしまいう問題点がある。

そこで、限られた表示面積で有益な可視化結果を得るために、次元削減手法を適用し本質的な情報を保持したまま多次元データを低次元空間で表示する手法がよく用いられてきた。しかし、次元削減手法を適用した手法では、縦軸と横軸に直感的に理解可能な数値を表現できないことが多い。この問題を解決するために、長崎らは、多次元データを特定の次元でクラスタリングして階層型データとして可視化する手法、およびその次元選択のために相関性の高い次元ペアを推薦する手法を提案した[15]。また Elmqvist らは、多次元の中から2次元を対話的に選択し、直交座標系のxy軸に割り当てて表示する Rolling the Dice を提案した[16]。さらに、Rolling the Dice による多次元データの表現形態を画像ブラウザに適用した例として、鄭らによる多次元データを有する大量画像群の一覧のための画像ブラウザ ImageCube がある[17]。本論文が提案する MusiCube に搭載された機能のうち、次元推薦機能は長崎らの次元推薦を拡張したものであり、次元選択に伴うアニメーション表示は Rolling the Dice における回転表示に類似するものである。

2.4 対話型進化計算と音楽推薦への応用

ユーザとのインタラクションによって人間の感性を最適化する手法として対話型進化計算 (Interactive Evolutionary Computation : IEC) がある。この手法は特に作曲やデザインなど、定量的な評価が難しいコンテンツの生成に多く適用されている。安藤らのCACIE では、クラシック音楽のアナリゼ手法に基づいて遺伝子表現し、作曲手法に基づいた生成プロセスで作曲する[18]。また、伊藤らは、対話型遺伝的アルゴリズムを用いてユーザの嗜好を反映する商品推薦システムを提案している[19]。

対話型進化計算を音楽推薦に適用した例として、Cho が、楽曲を音程の頻出回数でコーディングし、ユーザの5段階評価に基づいて、ユーザの嗜好に合った楽曲を推薦する手法を提案している[6]。また、Rho らは、楽曲をクエリに変換し、似たような楽曲を類似度順に提示する音楽検索システムを提案している。このシステムは、ユーザの満足度を高めるために、遺伝的アルゴリズム (Genetic Algorithm : GA) で最適化を行っている[7]。ただし、これらの手法は、進化計算によって得られたユーザの嗜好の傾向を表示しないため、ユーザがどのような特徴を持つ楽曲を好んでいるかということは読み取れない。楽曲特徴量が鍵となって、選曲の傾向が決まる場合、ユーザはその傾向に気づきにくいことがある。例えば、高音域の弱い楽曲を好む、不協和音が少ない楽曲を好むなどといった傾向に気づくユーザは少数であると思われる。そこで、ユーザ自身に選曲の傾向を気づかせるためにはインタラクションだけではなく可視化が有効であると期待できる。

進化計算における解探索の性能は、遺伝的演算手法や個体数、交

叉率、突然変異率などに大きく依存する。そのため、進化計算によって求められた解を可視化し、効率的な解探索を試みる研究もある。山代らの提案する進化計算の探索過程を可視化する手法では、探索過程において評価値空間における個体群の分布を可視化し、ユーザが探索の方向をインタラクティブに決定することによって、求める解を効率的に探索している[20]。この研究成果から、音楽推薦のための進化計算においても、個体群の可視化と対話操作を適用することで、より少ない評価回数で満足度の高い推薦結果を得ることが期待できる。MusiCube はこのような発想から生まれた手法である。

3. 提案手法

3.1 システム概要

本章では、楽曲ファイルから抽出した特徴量に基づ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 MusiCube を提案する。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。
- (7) ユーザにとって最適な楽曲特徴量を提示する。
- (8) プレイリストを作成する。

3.2 楽曲特徴量の検出

本手法では、数値解析ソフトウェア MATLAB の上に実装された楽曲分析パッケージ MIRtoolbox[21]を用いて、wave 形式の楽曲データから7種類の楽曲特徴量を抽出した (Table 1)。MusiCube では、これらの特徴量を同等に扱うため、楽曲特徴量 f を正規化した楽曲特徴量 f' を用いる。ここで、 $f' = (f - f_{min}) / (f_{max} - f_{min})$ とし、 f_{max} と f_{min} を楽曲特徴量の最大値と最小値とする。ただし、楽曲には次第に印象が変化するものや複数の印象を持つものがあり、楽曲全体の特徴量を一つに定めることは困難である。現段階では、楽曲における0分30秒から0分40秒の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。適用事例として用いた楽曲データベースにおいて、各楽曲のイントロの長さを計測した結果、平均の長さが28.5秒だったため、イントロが終了してAメロが始まった0分30秒から0分40秒の区間を対象とした。

Table 1 Music features.

特徴量	説明
RMS energy	音量の二乗平均の平方根
Low energy	弱音の割合
Tempo	テンポ
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	長和音と短和音の音量の差

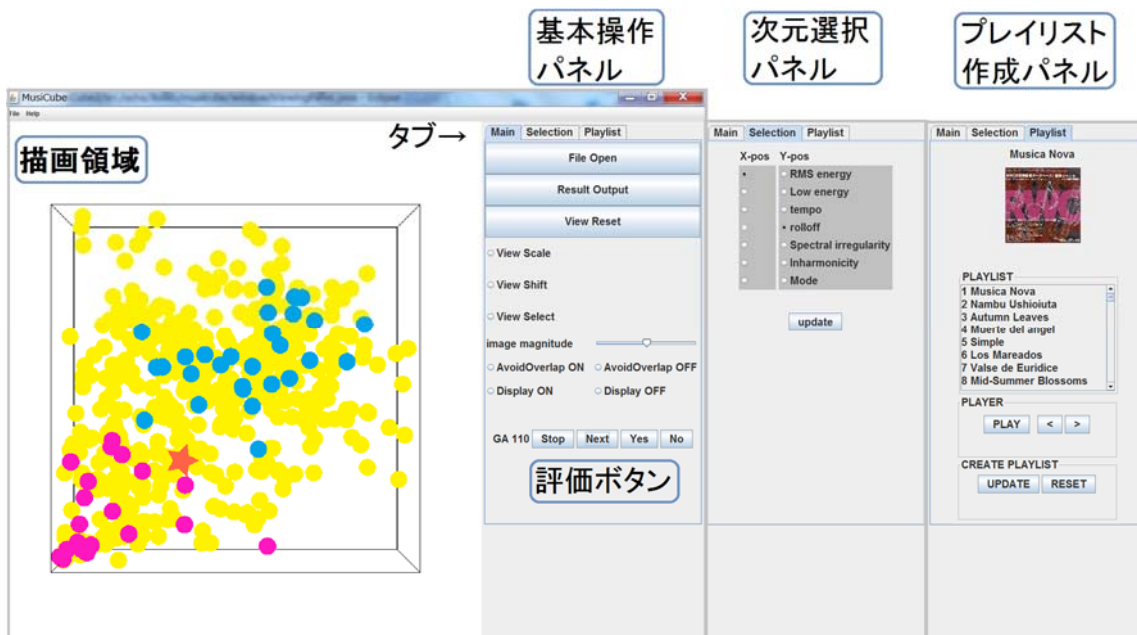


Fig. 1 Window design of MusiCube. Left side of the window displays a set of colored icons corresponding to tunes. Right side of the window has three tabs

3.3 MusiCube の表示機能と操作機能

本節では MusiCube を構成する各機能 (楽曲アイコン表示機能, 楽曲提示機能, プレイリスト作成機能) について説明する.

MusiCube の GUI 画面を Fig.1 に示す. 画面 (左側) の描画領域に楽曲群をカラーアイコンで表示し, 画面 (右側) のパネルに機能を切り替えるためのタブを表示する.

3.3.1 アイコンについて

MusiCube では, 楽曲を表現するアイコンの色と形状を, 以下のよう割り当てる.

- 黄色の丸 : 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.
- オレンジ色の星 : MusiCube によって提示された楽曲.
- 赤色の丸 : ユーザが目的に合っていると評価した楽曲.
- 青色の丸 : ユーザが目的に合っていないと評価した楽曲.

3.3.2 楽曲アイコン表示機能

Fig.1 に示す次元選択パネル上でユーザは, 各楽曲が有する楽曲特徴量の中から, 任意の2つの楽曲特徴量を選択する.

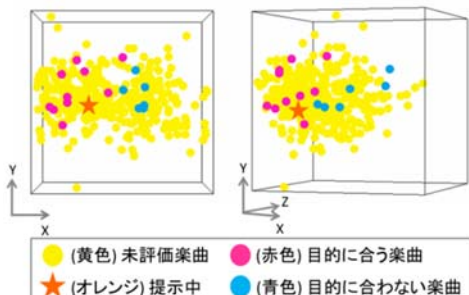


Fig. 2 (Left) Color icons corresponding tunes deployment. (Right) Color icons redeployment by rotation function.

MusiCube はそれらの2つの楽曲特徴量を, Fig.2(左)に示すように Scatterplot の xy 軸に割り当てる. この xy 平面上にて, 正規化した楽曲特徴量の値を座標値として, 各楽曲をアイコンで配置する.

また, xy 軸に割り当てる楽曲特徴量を別のものに差し替える際に, いったん z 軸に楽曲特徴量を割り当ててから 90 度回転表示することにより, シームレスに楽曲特徴量を差し替えながらアイコン群を再配置することができる. この回転操作は Rolling the Dice [16] における Scatterplot の回転表示に類似したものであり, Scatterplot の2軸を切り替える操作によってアニメーション表示で3次元空間を回転させる. Fig.2(右)は回転操作によるアニメーション表示の例である. このような回転操作によって, ユーザは次元間の関係を把握しやすくなり, 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容易に見ることができるようになる.

3.3.3 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. 対話型進化計算については3.4節で詳しく説明する. まず, 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を評価する. 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.

「MusiCube による楽曲提示」と「ユーザによる楽曲評価」を繰り返す, 提示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.

3.3.4 プレイリスト作成機能

MusiCube では、各楽曲に対する評価に基づいてプレイリストを生成する。立方体領域内においてユーザがマウスをクリックすると、MusiCube はクリックされ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。本手法ではユーザによって設定された xy 座標平面上におけるユークリッド距離 S を以下の式(1)で求め、一定範囲内（我々の実装では $S \leq 10$ ピクセル）に属する楽曲を選択楽曲としてプレイリストに追加する。ここで、 (x_{f1}, y_{f2}) はユーザによって設定された xy 座標平面上でのユーザがクリックした位置の座標値、 (x'_{f1}, y'_{f2}) は楽曲データの楽曲特徴量とし、 $f1$ と $f2$ はユーザによって選択された楽曲特徴量とする。

$$S = \sqrt{(x_{f1} - x'_{f1})^2 + (y_{f2} - y'_{f2})^2} \quad (1)$$

ただしプレイリストに追加できる楽曲は、赤色のアイコン（ユーザが目的に合っていると評価した楽曲）と黄色のアイコン（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）に対応する楽曲のみである。

3.4 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させた楽曲提示を行う。本手法では、任意の2次元の楽曲特徴量を選択し、それらの楽曲特徴量をFig.1に示す描画領域の xy 軸に割り当てている。しかし、遺伝的操作を行う際に、各楽曲が有する楽曲特徴量をそのまま遺伝子としてしまうと、次元数が多くなり探索性能が悪化してしまうおそれがあるため、対話型遺伝的アルゴリズムの遺伝子として、楽曲特徴量を主成分分析 (PCA) によって次元削減し、上位成分（第1～ n 成分）の値を採用する。そのため、実数値遺伝的アルゴリズム (Real-valued Genetic Algorithms) を採用し、遺伝子表現は、実数値ベクトルとする。Fig.3 に個体群のデータ仕様を示す。

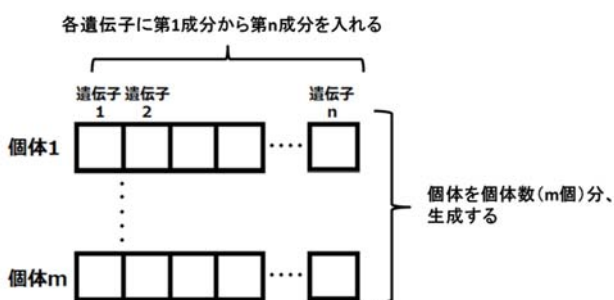


Fig. 3 Specification of population.

世代交代モデルには、Minimal Generation Gap[22]を採用する。ただし、本手法では、適合度を「目的に合っている」「目的に合っていない」の2値としているため、ユーザによって「目的に合っている」と判定された個体を中心に、2個の親個体（楽曲）を抽出して交叉するという処理を反復している。Fig.4 に対話型遺伝的アルゴリズムの処理の流れを、その詳細を(1)～(8)に示す。

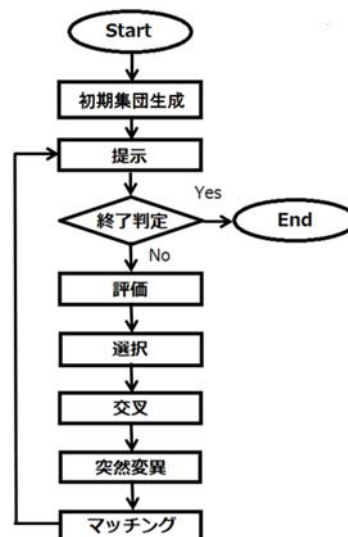


Fig.4 Process of interactive Generic Algorithm

- (1) 初期集団生成
MusiCube では、全楽曲の中から、あらかじめ定めた数の初期個体（楽曲）を選択する。初期個体として選択された個体（楽曲）の数を個体数（ m 個）と呼ぶ。
- (2) 提示
ユーザに対して個体（楽曲）群を提示する。
- (3) 評価
ユーザに提示個体（楽曲）を2値（目的に合っているか否か）で評価させる。MusiCube では、収束時間を速めるため、一度評価した個体（楽曲）が再提示された場合、自動的に評価値を入力する。
- (4) 選択
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（楽曲）とする。ここで、親個体（楽曲）の数は母集団の半数以上と定める。そのため、親個体（楽曲）の数が母集団の半数を満たさない場合、母集団の半数を満たすまで、目的に合っていないと評価された個体（楽曲）の中からランダムに個体（楽曲）を選び、親個体（楽曲）とする。
- (5) 交叉
MusiCube ではまず、(4)で得られた親個体（楽曲）のうち、総当たりで組み合わせを作る。次に、以下の式(2)によって、各組み合わせで2つの親個体（楽曲）間のユークリッド距離 d を算出し、 d が小さくなるような親個体（楽曲）の組み合わせを母集団の半数分、選ぶ。

$$d = \sqrt{\sum_{i=1}^n (\alpha_i - \beta_i)^2} \quad (2)$$

ここで、 α と β は2つの親個体（楽曲）を表し、それぞれ楽曲特徴量を主成分分析によって次元削減して算出さ

れた上位成分 (第1~n成分) の値とし, nは遺伝子の長さとする。

さらに, 選ばれた2つの親個体 (楽曲) の組み合わせで, BLX- α (ただし $\alpha = 0$) [22]に基づく交叉法を適用する。具体的には, Fig.5のように各遺伝子において2つの親個体 (楽曲) の値の範囲からランダムに値をとり, 2つの子個体の値とする。交叉は, 交叉率に従って個体に適用する。

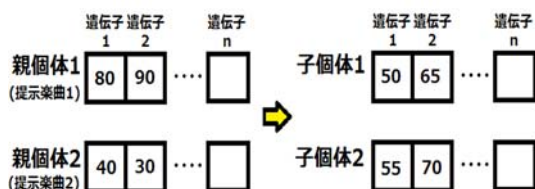


Fig. 5 Crossover

(6) 突然変異

個体群の多様性を維持するため, 一様突然変異を適用する。具体的には, 突然変異率にしたがって Fig.6のように各子個体の遺伝子をランダムに変化させる。

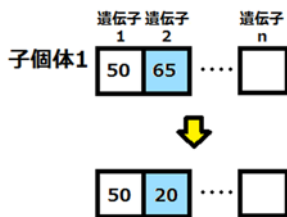


Fig.6 Mutation

(7) マッチング

データベースに収録されている楽曲から得られる個体 (楽曲) を(6)で得られた子個体とマッチングさせ, データベースから次世代の個体 (楽曲) となるものを選択する。具体的には, 以下の式(3)によって, (6)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。次世代に残す個体の数は, あらかじめ定めた個体数 (m個) となる。

$$d = \sqrt{\sum_{i=1}^n (f_i - p_i)^2} \quad (3)$$

ここで, fは遺伝的操作で得られた楽曲特徴量, pは楽曲データベースに収録されている楽曲データの特徴量を主成分分析によって次元削減して算出された上位成分 (第1~n成分) の値, nは遺伝子の長さとする。

(8) 終了判定

ユーザが求める個体 (楽曲) が得られれば操作を終了する。そうでなければ, (2)から(7)まで繰り返す。

3.5 xy 軸に割り当てる最適な楽曲特徴量の提示

MusiCube では, 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自動的に選択することができる。まず, 特定の楽曲特徴量を xy 軸に割り当て, その表示領域内を $N * N$ 個の矩形領域に分割する。次に, 各矩形領域内において赤色のアイコンの数と赤色以外のアイコンの数を算出し, 以下の式(4)よりエントロピーの合計を算出する。

$$E_{sum} = - \sum_{i=1}^{N^2} (p_{ri} \log p_{ri} + p_{qi} \log p_{qi}) \quad (4)$$

$$p_{ri} = \frac{r_i}{r_i + q_i} \quad (r_i + q_i) \neq 0$$

$$p_{qi} = \frac{q_i}{r_i + q_i} \quad (r_i + q_i) \neq 0$$

ここで, i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, 全楽曲特徴量を総当たりで xy 軸に割り当て各エントロピーを算出し, エントロピーが最も小さい値となった楽曲特徴量をユーザにとって最適な楽曲特徴量とする。このようにして選択された最適な楽曲特徴量を, Fig.1の次元選択パネルにおいて特定のカラーフォントで提示する。

4. 実験

4.1 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 2.4GHz, RAM 2.0GB) および Windows7 の上で実行した。また, システムの評価にあたり, 楽曲データ 500 曲を用意した。なお, 使用した楽曲データは, RWC 研究用音楽データベース (ポピュラー音楽, ジャズ音楽, 音楽ジャンル) に収録されたものなどで, 11 ジャンル (ポップス, ロック, ダンス, ジャズ, ラテン, ワールド, クラシック, ブラスバンド, 声楽, 邦楽, ア・カペラ) に及ぶ。

本実験では, 大学で情報科学を専攻している学生 10 人を対象に, 「MusiCube を使って, あるカフェをイメージして, そのカフェで再生したい楽曲を選ぶ」というテーマで実験を行い, 実験終了後にアンケートを実施した。実験内容は以下のとおりである。

- (1) MusiCube によって提示された楽曲を聴く。
- (2) 提示楽曲に対して, カフェの BGM として合うか否かを評価する。
- (3) (1)~(2) を繰り返す。
- (4) プレイリストを作成する。

なお本実験の予備実験として, 交叉率(c)と突然変異率(m)を変化させて, 各世代において目的に合っている楽曲の数を検証したところ, 交叉率 $c=0.9$, 突然変異率 $m=0.1$ とした場合に, 目的に合っている楽曲がより多く提示された。この結果より, 本実験では, 個体数 10, 交叉率 0.9, 突然変異率 0.1 と設定した。さらに,

個体遺伝子長(n)を変化させて、収束する速度を検証したところ、 $n=4$ とした時に、より速く最適解へ収束した。この結果より、遺伝子長4と設定した。

4.2 実験結果と考察

4.2.1 ユーザの嗜好の傾向

被験者4名の実験結果を Fig.7 に示す。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。Fig.7(A)と Fig.7(B)の被験者らは、 x 軸に割り当てられた楽曲特徴量 $RMS\ energy$ の値が小さく、 y 軸に割り当てられている楽曲特徴量 $Roll\ off$ の値も小さい楽曲を目的に合っている楽曲として評価している。 $RMS\ energy$ は音量の二乗平均の平方根を表す楽曲特徴量である (3.2 節)。ロックやポップスなどで電子楽器が主導する楽曲の多くは、ほぼ一定の音量で曲が進むため $RMS\ energy$ の値が大きくなる。一方、アコースティック楽器で構成された楽曲は、演奏上の抑揚や楽器の特性をもった音響的变化をもって曲が進み、相対的に $RMS\ energy$ の値が小さくなる。また、 $Roll\ off$ は、85%を占める低音域の値を表す楽曲特徴量である (3.2 節)。そのため例えば、高音域の倍音を多く含む金属楽器音 (例：タンバリン、シンバル) で構成された楽曲は $Roll\ off$ の値が高くなる。このことから、Fig.7(A)と Fig.7(B)の被験者らはカフェの BGM に対して類似した嗜好を持っており、両名ともアコースティック楽器で構成され、高音域の倍音成分が小さい楽曲を選んでいることが示唆される。

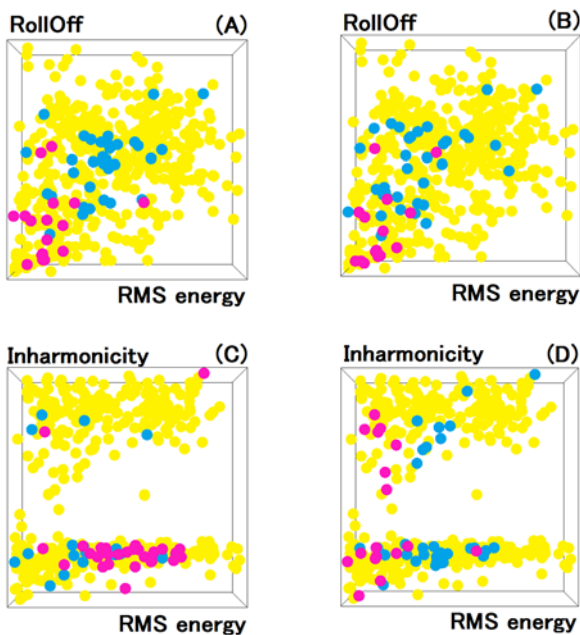


Fig. 7 Examples.

一方、Fig.7(C)の被験者は、 x 軸に割り当てられた楽曲特徴量 $RMS\ energy$ の値が比較的大きく、 y 軸に割り当てられている楽曲特徴量 $Inharmonicity$ の値が小さい楽曲を目的に合っている楽曲として評価している。 $Inharmonicity$ は根音に従っていない音の量を表す楽曲特徴量である (3.2 節)。そのため、不協和音が多い楽曲は $Inharmonicity$ の値が大きくなり、不協和音が少ない楽曲は

$Inharmonicity$ の値が小さくなる。このことから、Fig.7(C)の被験者はカフェの BGM として、ロックやポップスなどの電子楽器で構成され、かつ不協和音の少ないシンプルな楽曲を選んでいることが示唆される。この結果より、Fig.7(C)の被験者は Fig.7(A)や Fig.7(B)の被験者らと異なる嗜好を有すると考えられる。

また、Fig.7(C)と Fig.7(D)の被験者らは、 xy 軸に同じ楽曲特徴量が割り当てられているが、Fig.7(C)の被験者は x 軸に割り当てられた楽曲特徴量 $RMS\ energy$ の値が大きく、 y 軸に割り当てられた楽曲特徴量 $Inharmonicity$ の値が小さい楽曲を、Fig.7(D)の被験者は x 軸に割り当てられた楽曲特徴量 $RMS\ energy$ の値が小さく、 y 軸に割り当てられた楽曲特徴量 $Inharmonicity$ の値が小さい楽曲や大きい楽曲を目的に合っている楽曲として評価している。このことから、Fig.7(C)の被験者は、電子楽器で構成されていて、不協和音が少ないシンプルな楽曲を選んだのに対して、Fig.7(D)の被験者は、「アコースティック楽器で構成されていて、不協和音が少ないシンプルな楽曲」と「アコースティック楽器で構成されていて、不協和音が多い (例えばジャズ風の) 楽曲」という2種類の楽曲をカフェに合う音楽として選んでいることが示唆される。この結果より、Fig.7(C)と Fig.7(D)の被験者らは異なる嗜好を持っていることが考えられる。

また被験者に「その時の気分に対応しい楽曲をユーザの嗜好を反映させて選曲する場合、Fig.7に示すような可視化結果は効果的であるか」というアンケートを実施したところ、5段階評価で最大値5、平均値3.9、最小値3という結果を得た。

これらの結果より、「カフェに合う曲」というような特定の目的に対応しい楽曲を選ぶ場合において、MusiCubeには可視化技術の搭載によってユーザに自身の嗜好がどのような特徴量に起因しやすいかを気づかせる効果があることがわかる。また、ユーザの嗜好を学習するアルゴリズムとして対話型進化計算を用いることで、Fig.7(D)において2種類の楽曲を選んだのが観察されるように、「嗜好の多峰性」にも対応できていることがわかる。

4.2.2 最適な楽曲特徴量の提示

各被験者の結果において、楽曲特徴量の任意ペアを総当たりで xy 軸に割り当て、赤色のアイコンのエントロピーを算出した。MusiCubeではエントロピーを赤色のアイコン（目的に合っている楽曲）群が立方体領域内で局所集中して表示されているかを表す指標値として扱っており、実際にこの値が小さいほうが赤色のアイコンが局所集中する傾向にあることを目視確認している。Fig.8にて、10人の被験者の赤色のアイコンのエントロピーの最小値、平均値、最大値をグラフで示す。なおグラフの横軸は被験者を表し、縦軸はエントロピーを表す。この結果より、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、エントロピーを最小にする楽曲特徴量ペアを xy 軸に割り当てたほうが、赤色のアイコンの密度が高くなることがわかった。このことから、MusiCubeにおいて最適な楽曲特徴量を提示する機能が有効であると言える。

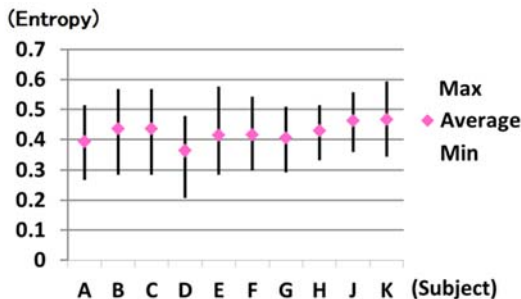


Fig. 8 Entropy. Minimum, average, and maximum E_{sum} values from the results of 10 subjects (A-K).

また, MusiCube を使って可視化結果を表示しながら学習を進め, その学習結果を用いてプレイリストを作成する過程で, ユーザがどのくらいの頻度で xy 軸に割り当てる楽曲特徴量を切り替えたかについて検証する. Fig.9 に 2 人の被験者の結果を示す. なおグラフの横軸は時間経過を表し, 縦軸は被験者を表す.

被験者 A は, 対話型進化計算によって嗜好を学習している間に 2 回, xy 軸に割り当てる楽曲特徴量を切り替えていた. 被験者 A に対して, 最適な楽曲特徴量を提示する機能が有効であるかとヒアリングしたところ, 評価した楽曲が増えてきた段階で, 赤色のアイコンの密度が高くなるような楽曲特徴量のペアをシステムに提案してもらうことで, 自身では気づきにくい嗜好の傾向を観察しながら, 一貫性のある選曲ができたというコメントをもらった. 一方, 被験者 B は, 嗜好を学習し始める前に 1 回, 学習している間に 3 回, xy 軸に割り当てる楽曲特徴量を切り替えていた. 被験者 B からは, 自身の嗜好が当初予想していなかった楽曲特徴量に起因していることに気づいたというコメントをもらった. 被験者 B は, 学習開始前に, 自身の嗜好に起因する楽曲特徴量を予想し, x 軸に RMS energy を, y 軸に Roll off を割り当てて, 学習を開始し, 学習時に複数回 xy 軸に割り当てる楽曲特徴量を切り替え, x 軸には Inharmonicity, y 軸には Roll off が割り当てた結果を用いてプレイリスト作成していた. この結果から, ユーザが任意の楽曲特徴量を xy 軸に割り当てることで, 自身の傾向を予想しながら学習を進めることができ, さらに, システムで提示する最適な楽曲特徴量を xy 軸に割り当てることで, ユーザ自身では気づきにくい嗜好の傾向を提示できることがわかる.

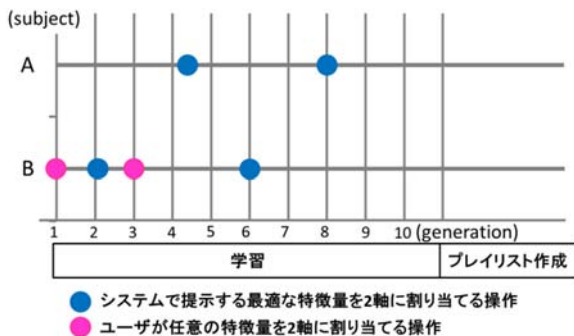


Fig. 9 Number of the selecting dimension.

ところで, 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づかせるインタフェースとなっている. 一方で, 楽曲特徴量の次元削減によってアイコン群の位置を決定した方が有用な場合があるかもしれない. そこで, 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に, 線形判別分析(Linear Discriminant Analysis : LDA) [23][24]を適用し, 本手法の結果と比較した.

Fig.10 に 2 人の被験者の結果において, 「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 xy 軸に割り当てて表示した場合 (Fig.10(A) と Fig.10(B)) と線形判別分析を適用し楽曲群を表示した場合 (Fig.10(C) と Fig.10(D)) のそれぞれの可視化結果を示す.

4.2.1 節でも論じたように, Fig.10(A) と Fig.10(B) は, xy 軸に同じ楽曲特徴量が割り当てられていて, 赤色のアイコン (目的に合っている楽曲) の配置も似ていることから, 被験者らが類似した嗜好を持っていることが読み取れる. しかし, Fig.10(A) の被験者の結果に線形判別分析を適用し楽曲群を表示した Fig.10(C) と, Fig.10(B) の被験者の結果に線形判別分析を適用し楽曲群を表示した Fig.10(D) を比較しても, 被験者らが類似した嗜好を持っているということは読み取れない. つまり, 本手法のように「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 xy 軸に割り当てて表示した場合, ユーザは自身の嗜好がどのような特徴に起因しやすいかを把握することが可能になり, さらに可視化結果を比較することで被験者らの嗜好の差を分析することができる. しかし, 線形判別分析 (LDA) によって楽曲群を表示した場合, 嗜好が類似するはずの被験者間でも可視化結果に大きな差異が生じてしまうことから, ユーザ間の嗜好の類似度を視覚的に判断することが困難になる. この点からも MusiCube に利があるといえる.

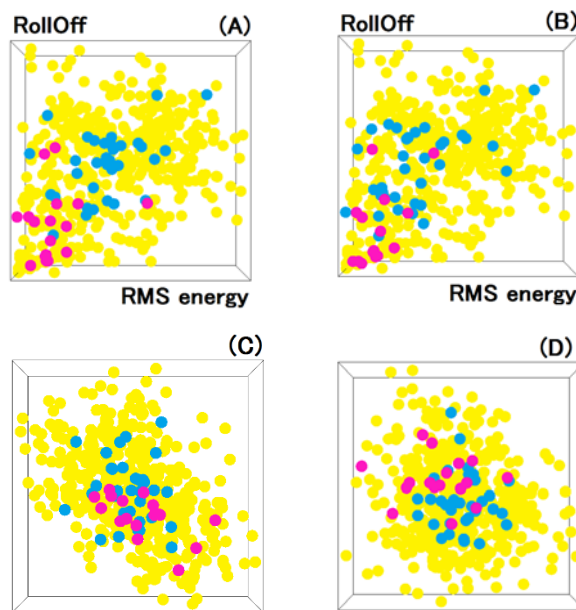


Fig. 10 Example of MusiCube and LDA.

また、我々は本手法のようにユーザの嗜好に大きな影響を与える特徴を可視化し、それに基づいてユーザがプレイリストを作成することで、より満足度の高いプレイリストを作成できると考えている。今後の課題として、プレイリスト作成結果に関するユーザの満足度を比較するための追加実験を実施したい。

4.2.3 推薦精度の比較

MusiCube の推薦精度を評価するにあたり、まず、対話型進化計算における推薦精度について検証する。音楽推薦の手法として、協調フィルタリングとユーザの嗜好情報のモデリングによる内容フィルタリングが挙げられるが、両者とも推薦の満足度を向上するためには時間を要する。協調フィルタリングでは、システムを利用するユーザ数が増えないと、推薦の満足度が上がらないため、時間がかかる。また、ユーザの嗜好情報のモデリングによる内容フィルタリングでは、あらかじめユーザが好む楽曲を一定数集めないと、嗜好の傾向をモデリングすることができない。そのため、1曲ずつ聞くことにより徐々に推薦精度があがる進化的な効果はなく、推薦を開始するまでに一定の時間がかかる。一方、MusiCube では、進化計算を採用することで、ユーザが1人しかいなくても推薦の満足度を上げることができ、また、1曲ずつ聞いていくうちに徐々に推薦の満足度を上げることができる。その結果、比較的早期に満足度が高い推薦結果が得られる可能性がある。そこで、評価楽曲数の増加に伴い、推薦精度（被験者10人の平均）がどのように変化したかについて、Fig.11に示す。ここで、推薦精度は、各世代（10個体）における目的に合っている楽曲の割合とする。縦軸は、1世代（10曲）中、目的に合っている楽曲と評価した楽曲の割合を、横軸は、世代を示している。評価楽曲数が少ない第1世代では、0.423であるのに対して、評価楽曲数が多くなる第10世代では0.925となった。この結果より、対話型進化計算を適用することで、評価楽曲数の増加に伴い、各ユーザにとって最適な楽曲を精度よく提示できていると言える。

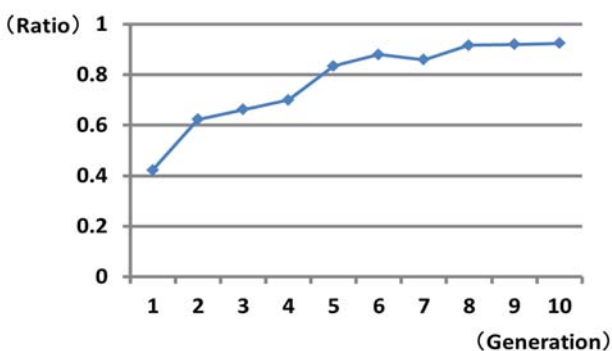


Fig. 11 Accuracy of Randomly recommendation and MusiCube .

次に、ユーザが目的に合っていると思う楽曲が提示されるまでのステップ数を検証する。MusiCube では、1世代を10曲とし、収束時間を速めるため、一度評価した楽曲が再び提示された場合、自動的に評価値を入力するようにしている。Fig.12に、Fig.10(A)の被験者のステップ数を示す。縦軸は、1世代（10曲）中の実際

に評価した楽曲と前回の評価を使った楽曲の割合を、横軸は、世代を示している。この結果より、ユーザは14世代以降、40曲程度評価した時点で、目的に合っている楽曲の割合が9割程度になっていることがわかる。

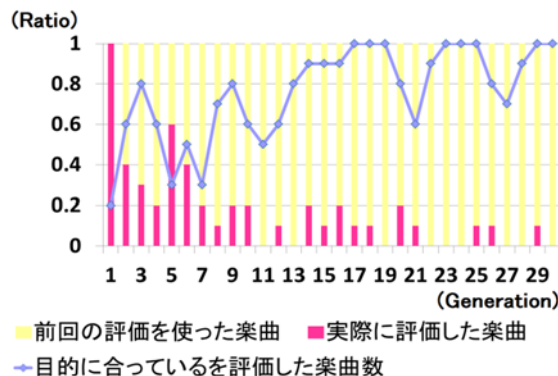


Fig. 12 Number of steps from the result of one subject.

また、対話型遺伝的アルゴリズムにおける1世代の処理時間について検証する。「選択」から「マッチング」までの処理時間を計測したところ、10回の平均が31.6ミリ秒であった。我々の主観では1世代の処理時間が1秒以内であれば対話的操作に十分であると考えているが、1世代の個体数が少ないことでMusiCubeの処理時間は対話的操作に十分なものとなっている。

さらに、MusiCubeのプレイリスト作成機能を用いて作成したプレイリストの推薦精度を検証する。MusiCubeでは3.3.4節で説明したとおり、立方体領域内で赤色のアイコン（目的に合っている楽曲）群の近くに配置されている楽曲を選択しプレイリストに追加することができる。プレイリストにおける推薦精度は、作成したプレイリストを使って被験者に楽曲を聴いてもらい、プレイリストに含まれる全曲のうち目的に合う楽曲の割合とする。ユーザには、可視化結果を表示しながら学習を進め、プレイリストを作成した場合と、可視化結果を表示せずに学習を進め、プレイリストを作成した場合の2通りを試してもらった。可視化結果を表示した場合のプレイリストにおける推薦精度は、最大値0.8、平均値0.64、最小値0.33、可視化結果を非表示にした場合のプレイリストにおける推薦精度は、最大値0.87、平均値0.58、最小値0.21となった。可視化結果を表示したほうが比較的高い満足度となり、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られる。

ユーザテスト後に可視化結果を表示しながら学習を進める場合と可視化結果を非表示にして学習を進める場合はどちらがよいかと聞いたところ、カフェに合う楽曲を選ぶ時のように、選曲に一貫性があることが重要な時、可視化はそのガイドラインとして非常に有益だと感じたという意見が多かった。一方、可視化結果が選曲における先入観になってしまうことを実験前に懸念したが、その点はあまり影響がないという意見が多かった。

ユーザの音楽に対する嗜好は、特徴量空間において多峰性を示すことがあるが、対話型進化計算がその複数の最適解領域を実際

に同時に探索しているのかはユーザにはわかりにくい。また、突然変異によって、ユーザの嗜好に全く合わない楽曲を提示されることもあるが、それもなぜ提示されたかユーザにはわかりにくい。MusiCube による可視化結果は、提示された楽曲や評価された楽曲の特徴量空間での位置を理解するのに有効であり、これを眺めながらのプレイリスト作成がユーザにとっての高い満足度につながっているものと考えられる。

続いて Fig.13 にて、10 人の被験者によって作成されたプレイリストにおいて、ジャンル別の楽曲の割合を 100%積み上げ棒グラフで示す。なおグラフの横軸は被験者を表し、縦軸はジャンル別の割合を表している。

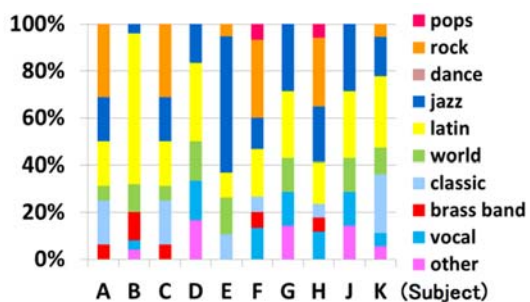


Fig. 13 Rate of genre in the 10 subjects' playlists (A-K).

この結果から、各プレイリストにはさまざまなジャンルの楽曲が含まれており、MusiCube がジャンルという枠を超えてユーザの目的に合う曲を選ぶのに有効であることがわかった。

4.2.4 GUI の使いやすさ

まず、評価のために用意した楽曲 500 曲の中からカフェに合う楽曲を 30 曲選ぶというテーマで、MusiCube を使わないでメタデータを見ながら選曲する場合と、MusiCube を使って学習時に可視化結果を表示した場合、MusiCube を使って学習時に可視化結果を非表示した場合で、それぞれ選曲にかかった時間を Fig.14 に示す。なお、グラフの横軸は時間を表し、縦軸は被験者を表している。学習時に可視化結果を表示した場合と非表示した場合は、プレイリストの作成にかかる時間に大きな差はなかったが、1 人の被験者を除いて、MusiCube を使った方が速くプレイリストを作成することができることがわかった。

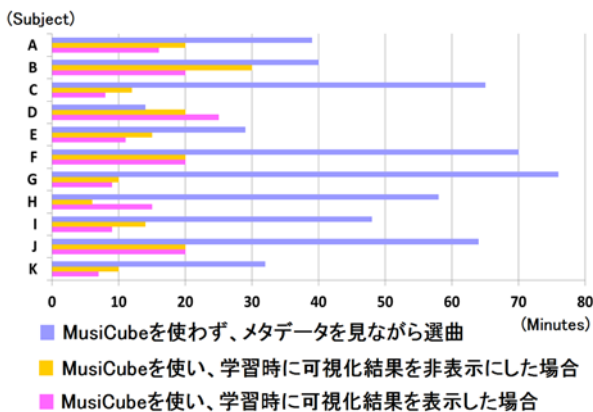


Fig. 14 Length of time it takes to make playlist from the results of 10 subjects (A-K).

最後に、被験者に「その時の目的に合う楽曲をユーザの嗜好を反映させて選曲する場合、MusiCube の GUI は効果的であるか」というアンケートを実施したところ、5 段階評価で最大値 5.0、平均値 4.0、最小値 3.0 という結果となった。

続いて、アンケートのフリーコメントを紹介する。

- 楽曲特徴量に加えて、ジャンルや歌詞などのメタデータを利用すると、さらに便利になるかもしれない。
- プレイリストを聴きながら、目的に合っていない楽曲だけを削除する、または GUI 画面 (左側) に表示されている立方体領域内でプレイリストに追加されなかった楽曲を聴き、目的に合っていたらプレイリストに追加するなど手で微調整を行う機能があれば、さらに直感的な選曲が可能になるかもしれない。
- 可視化結果がばらついていた場合、自分の定めていた「カフェの BGM」のイメージが曖昧だと認識できる。実際に BGM を選ぶ際、例えばもっとジャズっぽい雰囲気などと統一感のある雰囲気にできると思う。

我々は、これらのコメントを参考に MusiCube をより使いやすい楽曲提示インターフェースとして改良したいと考えている。

5. まとめと今後の課題

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ーフェース“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 で提示するインターフェースである。本手法を用いた評価実験結果では、MusiCube のような楽曲提示インターフェースを利用することにより、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。我々は、毎日カフェで働く人や毎日車を運転する人が毎日違う曲を聴きたい時などに、MusiCube を使って、その状況に合う新しい楽曲を選択すれば新鮮な気持ちで音楽を楽しむことができると考えている。MusiCube の特長として、以下の 2 点を挙げる。

- 多次元からなる楽曲特徴量を次元削減することなく、ユーザが任意の 2 次元の楽曲特徴量を対話的に選ぶことで、ユーザにとって重要な楽曲特徴量を気づかせることが可能になる。
- 対話型進化計算を用いてユーザの嗜好を学習し、その学習結果を可視化することで、大量の楽曲データの中から、ユーザが自身の嗜好に合った楽曲を直感的かつ効率的に探索することができる。

今後の課題として、以下の 4 点を挙げる。

まず 1 点目として、推薦精度を向上させるために楽曲特徴量をどのように採取するかについて引き続き検討していきたい。現時点では、楽曲における 0 分 30 秒から 0 分 40 秒の 10 秒間から得た楽曲特徴量をその楽曲の楽曲特徴量としている。今後は、楽曲全体を 10 秒ごとに分割しそれぞれから楽曲特徴量を抽出し、サ

ビの部分検出[25]などを適用することで最も適した楽曲特徴量をその楽曲の特徴量として扱いたい。また、被験者によっては可視化結果において赤色のアイコン（目的に合っている楽曲）群と青色のアイコン（目的に合っていない楽曲）群が混在してしまう結果となった。このような可視化結果となった原因について検討したところ、邦楽などの特殊な音階からなる楽曲の扱い方に問題があったと推測された。我々は3.2節で述べた楽曲特徴量に加え、非常に強い印象を与える特別なジャンルや歌詞などのメタデータを補足的に利用することで、MusiCube 本来の特徴を活かしつつ、より精度の高い推薦結果を提示できると期待している。

2点目の課題として、自身の嗜好情報と他者の嗜好情報を同時に表示できるような機能を追加し、さらに効率の良い楽曲提示を実現したい。

3.4節で述べた対話型進化計算を用いた楽曲提示では、抽出した楽曲特徴量に対して主成分分析を行い、上位成分の値を遺伝子として扱っている。3点目の課題として、このアルゴリズムにおいてユーザの嗜好の傾向に起因しやすい楽曲特徴量に重みをつけることで、より最適な収束結果を導けるかを試したい。

また、3.4節で述べた対話型進化計算を用いた評価では、収束時間を速めるため、一度評価した個体が再提示された場合、自動的に評価値を入力している。しかし、一度評価した楽曲を常に再提示したり、ある一定の割合で再提示することで、複数回の評価によって、被験者の評価を精密化したり、嗜好の変化に対応できる可能性がある。しかし、これらは、長期間に渡る実験が必要であり、実証に至っていない。4点目の課題として、一度評価した楽曲が再提示された時の評価手法を工夫することで、推薦精度を向上させることができるかについて引き続き検証していきたい。

以上について検討した上で、膨大な楽曲データを用いた実験を実施し、MusiCubeの有用性やスケーラビリティを再検証したい。

謝辞

本実験のデータとして、RWC 研究用音楽データベース（ポピュラー音楽、ジャズ音楽、音楽ジャンル）を使用した。本研究の一部は、日本学術振興会科学研究費補助金の助成に関するものです。

参 考 文 献

- 1) RWC-Music-Database. <http://staff.aist.go.jp/m.goto/RWC-MDB/>
- 2) 吉井和佳, 後藤真孝: 音楽推薦システム, 情報処理学会学会誌, 50(8) (2009) 751-755.
- 3) 松原正樹, 李軒, 斎藤博昭: 音楽への感性を開拓するメタ認知支援ツール, 日本認知科学会第27回大会 (2010) P3-44.
- 4) Hoashi, K., Matsumoto, K., Inoue, N.: Personalization of user profiles for content-based music retrieval based on relevance feedback, 11th ACM int'l Conference on Multimedia (2003) 110-119.
- 5) 梶克彦, 長尾確: 楽曲に対する多様な解釈を扱う音楽アノテーションシステム, 情報処理学会論文誌, 48(1) (2007) 258-273.
- 6) Cho, S.B.: Emotional Image and Musical Information Retrieval With Interactive Genetic Algorithm. Proc. IEEE 2004 (2004), 702-711.
- 7) Rho, S., Hwang, E., Kim, M.: Music Information Retrieval Using

- a GA-based relevance Feedback, International Conference on Multimedia and Ubiquitous Engineering (MUE '07) (2007).
- 8) Takagi, H.: Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation. In Proceedings of the IEEE, 89(9) (2001) 1275-1296.
- 9) Pampalk, E.: Islands of music: Analysis, organization, and visualization of music archives, Master's thesis, Vienna University of Technology (2001).
- 10) Newmayer, R., Dittenbach, M., Rauber, A.: PlaySOM and PocketSOMPlayer: alternative interfaces to large music collections, In Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05) (2005) 618-623.
- 11) Leitich, S., M. Tops, M.: Globe of Music Library Visualization Using GEOSOM, In Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07) (2007).
- 12) 浜脇修平, 石先広海, 帆足啓一郎, 滝嶋康弘: 楽曲の音響的特徴空間視覚化を利用した音楽検索システム, 情報処理学会音楽情報科学研究会 (2009) 29-34.
- 13) Stober, S., Nurnberger, A.: MusicGalaxy: A Multi-focus Zoomable Interface for Multi-facet Exploration of Music Collections, In Proceedings of 7th International Symposium on Computer Music Modeling and Retrieval (CMMR'10), 6684 (2010) 273-302.
- 14) Inselberg, A.: The Plane with Parallel Coordinates, The Visual Computer, 1(2) (1985) 69-91.
- 15) Nagasaki, A., Itoh, T., Ise, M., Miyashita, K.: A correlation-based hierarchical data visualization technique and its application to credit card fraud data, 1st International Workshop on Super Visualization (2008).
- 16) Elmqvist, N., Dragicevic, P., Fekete, J.: Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation, IEEE Transactions on Visualization and Computer Graphics, 14(6) (2008) 1141-1148.
- 17) Zheng, Y., Gomi, A., Itoh, T.: ImageCube: A Browser for Image Collections Associated with Multi-Dimensional Datasets, 15th International Conference on Information Visualization (IV2011) (2011) 12-17.
- 18) 安藤大地: 対話型GPを用いたクラシック音楽のための作曲支援システム, 芸術科学会論文誌, 4(2) (2005) 77-86.
- 19) 伊藤冬子, 廣安知之, 三木光範, 横内久猛: 対話型遺伝的アルゴリズムにおける嗜好の多峰性に対応可能な個体生成方法, 人工知能学会論文誌, 24(1) (2009) 127-135.
- 20) 山代大輔, 吉川大弘, 古橋武: 遺伝的アルゴリズムの解探索過程の可視化による遺伝的演算効果の把握と解探索の効率化, 情報処理学会論文誌, 48(SIG14(TOM 16)) (2007) 69-77.
- 21) MIRtoolbox. <https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox>.
- 22) 廣安知之, 三木光範, 福永隆宏: 遺伝的アルゴリズムにおける実数値ベクトル表現, 世代交代モデル, 母集団分割効果の検討, 同志社大学理工学研究報告 44(1) (2003-2004) 25-35.
- 23) Choo, J., Bohn, S., Park, H.: Two-stage framework for visualization of clustered high dimensional data. In IEEE Symposium on Visual Analytics Science and Technology (VAST 2009) (2009) 67-74.
- 24) Choo, J., Lee, H., Kihm, J., Park, H.: iVisClassifier: An Interactive Visual Analytics System for Classification Based on Supervised Dimension Reduction, In Proceedings of IEEE Symposium on Visual Analytics Science and Technology (2010) 27-34.
- 25) 後藤真孝: リアルタイム音楽情景記述システム: サビ区間検出法, 情報処理学会 音楽情報科学研究会 研究報告, 2002-MUS-47-6 (2002) 27-34.

