

コード進行の可視化を用いた楽曲検索・楽曲推薦システムの構築

甲斐彩有里 (指導教員：伊藤貴之)

1 はじめに

自分の好みの楽曲内容に類似した楽曲を探すには、楽曲を構成する三大要素である「メロディー・リズム・ハーモニー」を検索に用いる必要があるが、現在普及している楽曲検索システムの多くは「曲名・アーティスト・作曲家」などのキーワードをクエリに用いるのが一般的である。そこで本研究では、楽曲の進行を司るとともにメロディーの構成にも大きな影響を与えるコードに着目し、これを分析・可視化することで、楽曲検索・楽曲推薦が可能となるシステム「chord-vis」を構築した(図1)。本システムでは、まず大量の楽曲の分布をコード進行の類似度にもとづいて可視化し、そこからの絞り込みによって楽曲を検索・推薦できる。



図1: 「chord-vis」

2 関連研究

本章では、これまで提案されている楽曲の分析・可視化手法について紹介する。楽曲の特徴をコード進行から分析する研究例として、ヒットチャートランキング上位に入る楽曲の特徴分析 [1]、コード進行によるヒット曲予測システムの構築 [2] がある。以上にあげた手法はヒット曲のコード進行を分析することでヒット曲に共通する特徴を掴み、それをもとにヒット曲を予測することを目的としている。それに対して本研究は、楽曲の分析をヒット曲に限定せず楽曲間の類似度を可視化することで、楽曲内容に基づいた楽曲検索・

推薦が可能となるシステムの構築を目標としている。また、楽曲をコード進行をもとに分析し可視化する研究例として、コード進行に注目した J-POP 音楽の可視化 [3] がある。以上にあげた手法は、文字列が定型コード進行と一致した場合にコード進行を検出するため、類似性を柔軟に求めることが難しい。さらに通常のコードに3度ずつ音を堆積させた、楽曲のジャンルに影響を与える複雑な響きをする特殊なコードである「テンションコード」についての分析もされていない。それに対して本研究は、コード進行の類似をレーベンシュタイン距離を用いて求め、メタ情報に、テンションコードも加えることで、より柔軟なコード進行の検出・楽曲内容に基づいた分析を可能にした。

3 chord-vis の処理手順

本章では提案手法である chord-vis の処理手順について述べる。処理手順は、コード譜の取得、コード進行の解析、可視化用ファイルの生成、楽曲検索・楽曲推薦システムの構築、の4段階で構成される。各処理の詳細について、以下に論述する。

3.1 コード譜の取得

まず、各楽曲の「コード譜」と「楽曲名・アーティスト・作者」情報を抽出する。我々の実装では、10万を超える楽曲のコード譜が公開されているインターネット上のコード譜公開サイト [4] から情報を抽出している。ここから入手した HTML ファイルから「楽曲名・アーティスト・作者・コード進行」の掲載部分を文字列データとして抽出し、JSON ファイルに保存することで、各楽曲の「コード譜」と「楽曲名・アーティスト・作者」情報を抽出している。

表1: 我々の実装で採用している代表的な定型コード進行

名称	進行
王道進行	F,G,Em,Am
丸の内進行	FM7,E7,Am7,C7
小室進行	Am,F,G,C
小室マイナー進行	Am,Dm,G,Am
LetItBe 進行	C,G,Am,F
4156 進行	F,C,G,Am
6415 進行	Am,F,C,G
1625 進行	C,Am,Dm,G7
1451 進行	C,F,G,C
カノン進行	C,G,Am,Em,F,C,F,G

3.2 コード進行の解析

次に、3.1 節に示した処理で保存した JSON ファイルからコード進行を解析する。一般的に、各楽曲から抽出した楽曲コード譜は固有の調性(キー)を有する。

我々の実装では、調性に依存せずにコード進行を比較できるように、まず全ての楽曲の調性をCメジャースケールに移調する操作を適用する。続いて、各楽曲のコード進行を1節ごとに区切り、表1で示した10種類の代表的な定型コード進行とのレーベンシュタイン距離を算出する。これを繰り返すことで、各楽曲における定型コードの使用割合を算出できる。このコード譜の分析結果を新たなJSONファイルとして保存している。

3.3 可視化用ファイルの生成

続いて、楽曲間のコード進行類似度を散布図で、各楽曲ごとのコード進行の出現割合を帯グラフで、テンションコード使用頻度を棒グラフで表示するためのデータを生成する。テンションコードはsus2, sus4, dim, add9, 6, (b 5), (b 9), (b 13), aug, (# 9), (# 11)の11種類を採用する。我々の実装では散布図生成に多次元尺度構成法(MDS: Multi-Dimensional-Scaling)を用いている。多次元尺度構成法によって各楽曲の二次元座標を算出するために、各楽曲間の距離を計算し、楽曲の距離行列をCSVファイルとして保存する。さらに、各楽曲ごとのコード進行の出現割合とテンションコード使用回数をそれぞれ帯グラフと棒グラフによって可視化するためのJSONファイルを生成する。

3.4 楽曲検索・楽曲推薦システムの構築

我々はJavaScriptのフレームワーク「Next.js」を用いて提案システムを構築し、インターネット上でアクセスできるようにしている。楽曲検索・楽曲推薦システム「chord-vis」は<https://chord-vis.com>, または<https://chord-vis.vercel.app>にアクセスすることで誰でも閲覧が可能である。

4 データ表示機能とユーザインタフェース

本章では、3.4節で述べた「楽曲検索・楽曲推薦システム」が提供するデータ表示機能とユーザインタフェースについて論じる。

4.1 トップページ

トップページでは、大量の楽曲の分布によって構成された散布図・キーワード検索といった機能から、自分好みの楽曲を探すことができる。現時点での我々の実装では、楽曲散布図上に楽曲群のコード進行類似度分布が表示されている。この楽曲散布図上の点をマウスでホバーすることで、楽曲名・アーティスト・作曲者を確認することができる。この操作により、自分の好みの楽曲にメロディーが類似した楽曲を視覚的に探すことが可能である。また、「コード進行から探す」、「テンションコードから探す」機能では各コード進行・テンションコードの使用回数をランキング形式で閲覧することができる。

4.2 分析詳細ページ

現時点での我々の実装では、

- 楽曲分析詳細ページ
- アーティスト分析詳細ページ
- 作曲者分析詳細ページ

の3種類の分析詳細ページを提供する。

各分析詳細ページでは楽曲の「調性」、「最頻出のコード進行・テンションコード」、「コード進行の使用比率」、「テンションコードの使用回数」、「コード進行が類似している楽曲上位20曲」、「テンションコードが類似している楽曲上位20曲」に関する分析情報を詳しく閲覧できる。これらの機能により、ユーザーは楽曲やアーティスト・作曲者とコード進行の関係に対する知見を深めることができる。

5 可視化結果

現時点での我々の実装では、U-FRET[4]から抽出した計1000曲を対象に、楽曲間類似度の可視化と楽曲検索・推薦システムの構築を行なった。コード進行のランキングからは、特定のコード進行がどのような楽曲でよく使われているかを知ることができる。例えば、「小室進行」は、wowaka, 黒うさPなどのボーカロイドプロデューサーが上位にランキングしており、小室進行がボーカロイドでの作曲の際好まれて用いられる傾向があることがわかった。また、RADWIMPS, flumpool, miwaといったアーティストはsus4やadd9といったテンションコードを1曲に平均10回以上使用しているなど、テンションコードの傾向・作曲の特徴を知ることができる。

6 まとめと今後の展望

本研究では、楽曲の三大要素の一つである「メロディー」に大きな影響を与える「コード」を大量の楽曲を対象に分析・可視化し、そこからの絞り込みをすることで、楽曲検索・楽曲推薦が可能となるシステム「chord-vis」を構築した。また楽曲のジャンルに影響を与える特殊なコード「テンションコード」の分析を加えることで、より楽曲内容に基づいた分析を可能にした。本システムでは、多次元尺度構成法を用いて楽曲間のコード進行類似度を散布図上で表示した。これによってユーザーは自分の好きな楽曲やアーティストに似た曲を視覚的に見つけることができる。各楽曲におけるコード進行割合やテンションコード数をそれぞれ円グラフ、棒グラフで可視化することで、各楽曲のコードの特徴を視覚的に捉えることができる。今後の課題として、分析精度を向上させることで、現時点の実装では代表的なコード進行が全く見つからなかった楽曲を分析対象として含めたい。また、構成音が共通していて響きが似ている、俗に「代理コード」についても考慮したい実装をしたい。

参考文献

- [1] 横山真男, 斉藤勇也, ヒットチャートランキング上位に入る楽曲の特徴分析, 情報処理学会研究報告, Vol. 2015-MUS-106, No. 22, 2015.
- [2] 吉田隼輔, 木村優介, コード進行によるヒット曲予測システムの構築, 研究報告音声言語情報処理, Vol. 2021-SLP-137, No. 12, 2021
- [3] 上原美咲, 伊藤貴之, コード進行に注目したJ-POP音楽の可視化, 芸術科学会論文誌, Vol. 15, No. 4, pp. 177-184, 2016.
- [4] U-FRET <https://www.ufret.jp/>