

YouTube における楽曲のカバー動画の時系列可視化 Time-Series Visualization of Musical Cover Videos in YouTube

眞野 豊広

Toyohiro Mano

法政大学情報科学部コンピュータ科学科

E-mail: toyohiro.mano.6z@stu.hosei.ac.jp

Abstract

YouTube is an online video sharing platform that started service in 2005. Since any users can easily post their videos to YouTube, there are various genres of videos. One major genre is musical cover videos with various songs such as fan-made videos. There are many varieties of musical cover videos where, for example, users sing songs and play instruments such as the piano and the guitar. Since musical cover videos can increase the popularity of the original songs, it is a useful means of promoting the songs. Therefore, creators compose songs by aiming at facilitating the musical cover videos of their songs. However, it is difficult to check how such aims affect musical cover videos. Therefore, it will be useful if there is a tool that can analyze musical cover videos. This paper proposes a method for analyzing musical cover videos by automatically classifying and visualizing them in chronological order. We process the data by morphological analysis and visualize and classify it by using TF-IDFs, cosine similarities, and hierarchical clustering. To easily analyze clusters, we present a function for showing clusters' legends and videos' titles. To easily find clusters' features that are hard to find, we provide a function for changing the number of clusters.

1. はじめに

YouTube は、2005 年にサービスが開始された動画配信プラットフォームである。YouTube では、一般のユーザが簡単に自身の作成した動画を投稿でき、その手軽さから様々なジャンルの動画が存在している。その内の1つが既存の楽曲を使用し、新たに創作を行っているカバー動画であり、様々な種類がある。例えばユーザ自身が楽曲を歌った音声によるカバー動画(歌ってみた)や、楽器で楽曲を演奏した映像によるカバー動画(弾いてみた、演奏してみた)などである。

クリエイターにとって、カバー動画は自身の楽曲の拡散力となるため、カバー動画を多く作成してもらうための狙いを考え楽曲を作ることは、楽曲を流行させる上で有用な手段の1つといえる。しかし、実際にクリエイターの狙いがカバー動画を作られる際にどのように影響を与えているかを確認することは困難である。そこで、クリエイターが様々な楽曲のカバー動画の分析を行い、それぞれ

の楽曲における傾向の違いについて考察できるツールがあれば便利である。

本研究では、クリエイターが様々な楽曲のカバー動画について分析を行うことを目的として、YouTube におけるカバー動画を自動で分類し時系列による可視化を行う手法を提案する。YouTube の API から得られるテキストデータを形態素解析によって処理し、TF-IDF とコサイン類似度を使用して階層型クラスタリングによる分類、可視化を行う。凡例やタイトルの表示機能を提供することで、クラスタの特徴についての理解、分析を容易にする。可視化のクラスタ数を自由に変更する機能を提供し、クラスタ数が少ない場合に見逃ししやすいクラスタの特徴を発見しやすくする。実験では既存の楽曲を使用し、本研究の提案手法による可視化を行った。凡例やクラスタ数の変更機能などを使用することで、クラスタの詳細な分析が可能であることを示す。

2. 関連研究

動画プラットフォームにおけるカバー動画のような連鎖的な創作について可視化する研究は少ない。濱崎ら [1] はニコニコ動画で発生する連鎖的な協調的創作活動について、ネットワークによる可視化・分析を行った。ニコニコ動画では、VOCALOID と呼ばれる音声合成ソフトを利用した楽曲が数多く投稿されている。VOCALOID を使用した楽曲周辺の文化では、実写の映像を使用したミュージックビデオ(MV)を使用されることは少なく、1~複数枚のイラストや CG モデルを使用しそれに歌詞を付けて編集した映像を MV として投稿されることが多い。よって VOCALOID を使用した楽曲はイラストや映像と密接な関係があり、ユーザが自由に使用可能なイラストを投稿し、そのイラストと VOCALOID を使用した楽曲を別のユーザが作成・投稿し、その楽曲を良いと思った別のユーザがその楽曲と別の画像・映像などを組み合わせ、新たな動画を作成するといったような連鎖的な創作活動が発生する。この研究ではそれらの動画を1つのノードとし、このような協調的創作活動にどのような特徴があるのかを分析するために、ネットワークによる可視化を行った。

3. 準備

3.1. TF-IDF

本研究では、テキストデータをベクトル化する際に TF-IDF を使用する。TF-IDF は単語の出現頻度を表す TF

と珍しい単語に対して重みを付ける逆文書頻度を表す IDF の積である。文書を d 、単語を t としたときの TF と IDF はそれぞれ以下のように定義される。

$$tf(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

$$idf(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

ここで、 $f_{t,d}$ は文書 d に含まれる単語 t の出現頻度、 N は全文書数、 $|\{d \in D : t \in d\}|$ は単語 t の出現する文書 d の数を表す。よって、TF-IDF は以下のように表せる。

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D)$$

3.2. コサイン類似度

コサイン類似度とは、2 つのベクトルの類似度を計算する手法であり、ベクトル \vec{a}, \vec{b} を用いて以下のように表せる。

$$\cos(\vec{a}, \vec{b}) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} \quad (-1 \leq \cos(\vec{a}, \vec{b}) \leq 1)$$

コサイン類似度が 1 に近いほど類似度は高く、-1 に近いほど類似度は低くなる。

3.3. 階層型クラスタリング

階層型クラスタリングは、クラスタを階層的に形成するクラスタリング手法である。図で示す際には、図 1 のような樹形図で表すことができる。階層型クラスタリングでは、樹形図の縦軸である閾値を変化させることで任意のクラスタ数によるクラスタリングが可能であるため、クラスタ数を事前に決める必要がないという点において k-means などのクラスタリング手法とは異なる。図 1 において、閾値が 2.0 のとき、クラスタは {1,2,3,4,6,7,9}, {0,5,8,10} の 2 つであるが、閾値が 0.5 のとき、{4,9}, {1,2,3,6,7}, {0,10}, {5,8} の 4 つのクラスタに分けることができる。

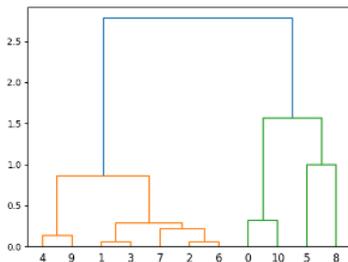


図 1 階層型クラスタリングの例

4. 提案手法

本研究では、YouTube 上に存在しているカバー動画のデータを収集したのち、それを処理することでそれぞれのカバー動画を自動で分類し可視化する手法を提案する。

4.1. テキストデータ収集

カバー動画の分類をする際に使用するデータとして、テキストデータを収集する。YouTube 上に存在する動画のタイトル・概要欄・コメントなどの情報は、YouTube の API によって容易に入手できるためである。ただし、API は 1 日に取得できるデータ量が指定されており、全

てのコメントを収集することは現実的でない。そのため今回はカバー動画のタイトルと概要欄を分類に使用した。

4.2. テキストデータ処理

テキストデータは収集した直後の無加工の状態では分類に使用できないため、大きく分けて 2 つの処理を行う。具体的には、不要な文字の削除とテキストデータの形態素解析を行う。不要な文字の削除では、まず、英語の文字を全て小文字に統一する。これにより、例えば「I Love YOU」という文字列がある場合「i love you」に置換される。次に URL を削除し、記号を削除する。次にテキストデータを Unicode 正規化する。これを行うことで、数字・アルファベットを半角に、半角カタカナを全角に変換できる。最後に数字と残りの記号、分類する楽曲のアーティスト名とタイトルを削除する。

続いて、テキストの形態素解析を行う。形態素解析とは文章をそれぞれの形態素に分解することである。例えば「私はパン派です」という文がある場合は「私・は・パン・派・です」というように 5 つの語に分解できる。このように分解したのち、名詞と動詞のみを残し分類に使用する。最後に自然言語処理において不要な語であるストップワードを削除する。具体的には、こそあど言葉や YouTube 上の動画に共通して書かれやすいチャンネル登録、高評価といった単語などを手動で選択し削除した。

4.3. テキストデータ分類

処理を行ったテキストデータを格納したリストを利用してカバー動画の分類を行う。まず、リストに対して TF-IDF を使用してテキストデータをベクトル化する。それぞれの単語の TF-IDF 値が高いほど、その単語は分類において重要度が高いということになる。続いて、コサイン類似度を使用してそれぞれの動画の類似度を計算する。ただし、コサイン類似度の値には負の数が含まれるため、1 からコサイン類似度の値を引いた値を使用する。その後、計算したコサイン類似度を使用して階層型クラスタリングを行う。

4.4. 可視化

階層型クラスタリングから任意のクラスタ数を設定し可視化を行う。可視化には階層型クラスタリングによる樹形図と時系列による可視化が可能な ThemeRiver [2] の 2 つを使用する。図 2 は階層型クラスタリングによる樹形図と ThemeRiver を使用した可視化結果である。ThemeRiver は 0 を中心とし、その上下に描画する積み上げ面グラフである。積み上げ面グラフでは、それぞれのカテゴリの変化と全体としての数値変化を時系列によって容易に確認できる。この手法は楽曲の流行を扱う上で適切であると考えた。本研究では、ThemeRiver の縦軸をそれぞれの分類の動画数、横軸を 1 週間ごとの時間として可視化を行う。また、ThemeRiver では凡例として各クラスタにつき 6 つの単語を表示する。この単語は TF-IDF 値が 0 でない単語について、それぞれ存在する文書数を各クラスタでカウントし、その文書数の上位 6 単語を選出している。凡例の表示によって、各クラスタがどのような特徴を持っているかを確認することが容易となる。

樹形図は、ユーザがクラスタ数を変更した際にそのクラスタ数に応じてそれぞれの要素の色が変化する。この色は各クラスタにおいて ThemeRiver で使用されている色と一致しているため、各クラスタを樹形図と ThemeRiver の2つで対応させて見る事が可能である。

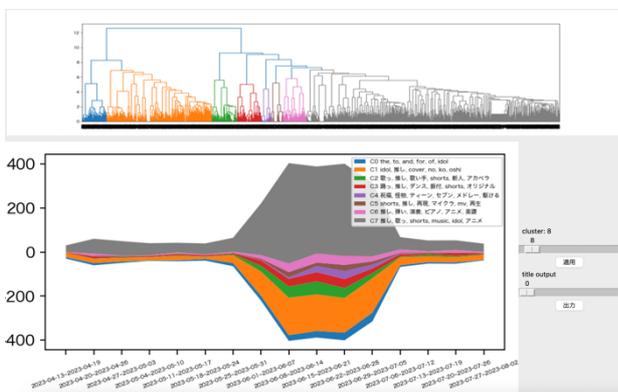


図2 楽曲「アイドル」のカバー動画の可視化結果

5. 実装

実装には Python を使用した。YouTube のテキストデータを収集するために YouTube Data API を使用した。テキストデータから絵文字を削除する際に demoji を使用し、形態素解析に janome を使用した。TF-IDF の計算には scikit-learn を使用し、コサイン類似度の計算と階層型クラスタリングには scipy を使用した。階層型クラスタリングの樹形図と ThemeRiver の可視化には matplotlib を使用した。GUI は Tkinter を使用し作成した。

GUI では、階層型クラスタリングに使用した樹形図と ThemeRiver を確認することができる。実行時にはクラスタ数が 5 のときの実行結果のみが表示されるが、画面右にある 1 つ目のスライダの数値を変化させボタンを押すことでグラフが再描画され、ユーザが設定したクラスタ数によるクラスタリング結果が確認できる。2 つ目のスライダでは、それぞれのクラスタに含まれるカバー動画のタイトルを別ウィンドウで表示する機能が使用できる。これにより、任意のクラスタのカバー動画について詳細に調べることができる。

6. 実験

6.1. データセット

実験に使用したデータセットを表 1 に示す。データは原曲の MV が YouTube 上で公開された日から数週間アップロードされたカバー動画について収集したものである。1 つの動画につき、動画のタイトル、動画 ID、チャンネル ID、投稿日、概要欄の 5 つの情報を収集した。

表 1 データセット

楽曲名 / アーティスト	動画数	収集期間
アイドル / YOASOBI	4524	16 週間

6.2. 実験概要

上記のデータセットと実装した GUI を使用して可視化を行う。対象とする楽曲は YOASOBI の「アイドル」である。図 2 は「アイドル」について本研究の手法によって可視化した結果である。クラスタ数は 8 とした。「アイドル」は漫画が原作のアニメ「推しの子」においてオープニングテーマとして使用された。このアニメは 2023 年 4 月 12 日から 2023 年 6 月 28 日まで放送されており、図 2 の ThemeRiver と凡例を見ると、アニメ放送中に「アイドル」のカバー動画総数が増加し、アニメ終了と同時に動画総数が減少傾向にあることが読み取れる。また、この傾向は特定のクラスタのみから読みとれるものではなく、どのクラスタにおいてもこの期間に動画が最も多く含まれることが分かる。

紫色のクラスタは凡例に「祝福」や「怪物」など YOASOBI の他の楽曲名や「メドレー」といった単語が含まれていることから、一般のユーザが投稿した YOASOBI 楽曲メドレーなどの動画のクラスタであることが凡例から推測できる。このように、凡例を使用することでそれぞれのクラスタの特徴を推測できる。

実際にどのような動画が含まれているか確認するために、GUI 右側の 2 つ目のスクロールバーを使用し動画のタイトルを別ウィンドウによって表示する。図 3 は紫色のクラスタに含まれる動画のタイトルを実際に表示した結果の抜粋である。これを見ると、凡例から推測した内容と同じく、YOASOBI の楽曲メドレーが多く含まれていることが分かる。また、全く同じタイトルの動画が多く、YOASOBI の楽曲メドレーは少数のユーザが似たような動画を複数アップロードしていることが読み取れる。このようにタイトルを出力する別ウィンドウを使用することで、凡例を使用する場合からより詳しくそれぞれのクラスタについて調べることが可能である。



図 3 紫色のクラスタに含まれる動画タイトルの抜粋

階層型クラスタリングでは、クラスタ数を大きくすることでそれぞれのクラスタをより細かく分類することができるため、続いてクラスタ数を 17 にして可視化を行う。図 4 は「アイドル」についてクラスタ数 17 として可視化した結果である。図 4 は図 2 と比較してクラスタ数が多く、より細かく分類されていることが分かる。図 2 においてピンクのクラスタの凡例を見てみると、「演奏」「ピアノ」などの単語があることから、弾いてみたというジャンルのカバー動画のクラスタであることが分かる。このクラスタがより細かく分類されたのが図 4 の真ん中あたりに存在する青と水色の 2 つのクラスタである。青のクラスタの凡例を見ると、「演奏」「ピアノ」などの図 2 における凡例と同様の単語が表示されていることが

