

# 最小スパニングツリー問題を用いた音楽ストリーミングサービスでの楽曲サジェスチョンサービス

坂本 嵩

環境情報学部2年

71775236

ssakamoto21@keio.jp

## このレポートで扱う問題を選ぶに至った背景や動機

私は日頃から良く音楽を聴く。お気に入りの曲を繰り返し聴くのはもちろん、新しい曲を見つけるのも楽しい。自分の好みにあった新しい曲を手動で見つけるのは簡単ではないので、私はYouTubeプレイリストを頻繁に使う。動画を一つ再生すると、ユーザーの好みに合わせて動画のプレイリストを作成してくれるサービスだ。これは素晴らしく、大抵の場合ファン層の似通ったアーティストの曲をおすすめしてくれる。例えば、邦ロックバンドの「エレファントカシマシ」を聴いていると、おすすめには同じく邦ロックバンドである「フジファブリック」が入ってくる。どちらも素朴ながらも確かな歌詞とメロディーでファンを魅了するバンドで、ファン層の親和性は高い。他のストリーミングサービスでもプレイリストを作成する機能はあるが、どのように「スムーズな」曲の遷移を行っているのだろうか？非常に興味深いところである。そこで私は、共通するファン数や曲の類似度などから曲間の「距離」をうまく設定して「曲グラフ」を設計し、最小スパニングツリー問題を解けば「スムーズな」サジェスチョンルートを構築できるのではないかと仮説を立てた。

## 問題の表現

まず、楽曲を頂点とした全結合グラフを作成し、この時の辺の長さを楽曲間の「距離」とする。スムーズな曲遷移を行うためには、2つの楽曲の親和性を計算し、その逆数を距離とすればよい。そこで今回、2つの楽曲間の「親和性」を  $L_{\text{mutual}}$  とおいてみる。ここで、 $L_{\text{mutual}}$  は楽曲を2つとも「Like」したユーザーの数である。2つの曲に共通するファンが増えれば楽曲の親和性も高まるのではないか。しかし、これでは元々「Like」の数が多い人気な曲にバイアスがかかってしまう ( $L_{\text{mutual}}$  が高くなりやすい) ため、これを  $L_{\text{total}}$  で正規化する。ここで、 $L_{\text{total}}$  は2つの楽曲を「Like」したユーザーの合計数である。つまり、最終的な「親和性」は  $L_{\text{mutual}} / L_{\text{total}}$  となり、楽曲間の距離は親和性の逆数となるため、全結合グラフにおける距離は  $L_{\text{total}} / L_{\text{mutual}}$  となる。これは、2つの楽曲の「Like」に関する OR を AND で割った形でもある。そうして作成した「楽曲グラフ」を、最小スパニングツリー問題で解く。曲遷移をする際の移動コストの合計値を最小化するのが目的であるが、最短経路ではなく最小スパニングツリーを計算することで、ユーザーに複数の候補曲を提示することができるため、よりユーザーにとって自由度の高いサービスを提供することができる。

## アルゴリズムの詳細

今回は、この「楽曲グラフ」において最小スパニングツリー問題をプリムのアルゴリズムで実装し、解いた。ソースコードは <https://github.com/mosh-shu/saitekika/blob/master/prim.ipynb> にアップロードされている。

楽曲には『星空のディスタンス/THE ALFEE』、『メリーアン/THE ALFEE』、『赤いスイートピー/松田聖子』、『ファッションモンスター/きゃりーぱみゅぱみゅ』、『ポリリズム/PERFUME』、『名もなき

詩/Mr.Children』, 『honnoji/ZAZEN BOYS』, 『ポテトサラダ/ZAZEN BOYS』の8つを選んだ. 昭和世代に人気の曲, 若者に人気の曲, 全世代に人気の曲, あまり有名ではないが根強いコアなファンを持つバンドの曲をチョイスした. YouTubeでの高評価数をもとに  $L_{total}$ ,  $L_{mutual}$ を算出した.

## 実証

求めた全結合楽曲グラフは, 行列表記で以下ようになった. 一つの配列には, 辺の両頂点にある楽曲 (頂点の名称) とその距離が格納されている.

```
[['星空のディスタンス/THE ALFEE', 'メリーアン/THE ALFEE', 5.148333333333333],
 ['星空のディスタンス/THE ALFEE', '赤いスイートピー/松田聖子', 6.932666666666666],
 ['星空のディスタンス/THE ALFEE', 'ファッションモンスター/きゃりーぱみゅぱみゅ',
 18.875],
 ['星空のディスタンス/THE ALFEE', 'ポリリズム/PERFUME', 15.0],
 ['星空のディスタンス/THE ALFEE', '名もなき詩/Mr.Children', 3.375],
 ['星空のディスタンス/THE ALFEE', 'honnoji/ZAZEN BOYS', 14.699],
 ['星空のディスタンス/THE ALFEE', 'ポテトサラダ/ZAZEN BOYS', 25.464000000000002],
 ['メリーアン/THE ALFEE', '赤いスイートピー/松田聖子', 4.7476666666666665],
 ['メリーアン/THE ALFEE', 'ファッションモンスター/きゃりーぱみゅぱみゅ', 41.27],
 ['メリーアン/THE ALFEE', 'ポリリズム/PERFUME', 11.7225],
 ['メリーアン/THE ALFEE', '名もなき詩/Mr.Children', 8.178],
 ['メリーアン/THE ALFEE', 'honnoji/ZAZEN BOYS', 16.288],
 ['メリーアン/THE ALFEE', 'ポテトサラダ/ZAZEN BOYS', 20.59],
 ['赤いスイートピー/松田聖子', 'ファッションモンスター/きゃりーぱみゅぱみゅ',
 16.644222222222222],
 ['赤いスイートピー/松田聖子', 'ポリリズム/PERFUME', 3.5997500000000002],
 ['赤いスイートピー/松田聖子', '名もなき詩/Mr.Children', 3.035058823529412],
 ['赤いスイートピー/松田聖子', 'honnoji/ZAZEN BOYS', 13.497],
 ['赤いスイートピー/松田聖子', 'ポテトサラダ/ZAZEN BOYS', 23.06],
 ['ファッションモンスター/きゃりーぱみゅぱみゅ', 'ポリリズム/PERFUME', 10.6],
 ['ファッションモンスター/きゃりーぱみゅぱみゅ', '名もなき詩/Mr.Children',
 15.600000000000001],
 ['ファッションモンスター/きゃりーぱみゅぱみゅ', 'honnoji/ZAZEN BOYS', 71.8495],
 ['ファッションモンスター/きゃりーぱみゅぱみゅ', 'ポテトサラダ/ZAZEN BOYS', 141.732],
 ['ポリリズム/PERFUME', '名もなき詩/Mr.Children', 3.5],
 ['ポリリズム/PERFUME', 'honnoji/ZAZEN BOYS', 15.132666666666665],
 ['ポリリズム/PERFUME', 'ポテトサラダ/ZAZEN BOYS', 25.915],
 ['名もなき詩/Mr.Children', 'honnoji/ZAZEN BOYS', 9.8495],
 ['名もなき詩/Mr.Children', 'ポテトサラダ/ZAZEN BOYS', 22.165],
 ['honnoji/ZAZEN BOYS', 'ポテトサラダ/ZAZEN BOYS', 3.194705882352941]]
```

ここから上のプリムのアルゴリズムで求めた最小スパニングツリーは, 行列表記で以下ようになった.

```
[['星空のディスタンス/THE ALFEE', '名もなき詩/Mr.Children'], ['赤いスイートピー/松田聖子', '名もなき詩/Mr.Children'], ['ポリリズム/PERFUME', '名もなき詩/Mr.Children'],
 ['メリーアン/THE ALFEE', '赤いスイートピー/松田聖子'], ['名もなき詩/Mr.Children', 'honnoji/ZAZEN BOYS'], ['honnoji/ZAZEN BOYS', 'ポテトサラダ/ZAZEN BOYS'], ['ファッショ
ンモンスター/きゃりーぱみゅぱみゅ', 'ポリリズム/PERFUME']]
```

ここで、プリムのアルゴリズムの特徴である "最小スパニングツリーとして完成している部分グラフから伸ばしうる (部分グラフ以外の) 最短の辺を選ぶ" という点に着目すると, "星空のディスタンス" から伸ばしうる最短の辺は "名もなき詩" であり, この2曲からの伸ばしうる最短の曲は "赤いスイートピー" である. そしてこれらの曲から伸ばしうる最短の辺は... とチェックをすると, 上に示した最小スパニングツリーと一致することが分かる.

## 考察

以上のように, 楽曲で全結合グラフを作成し, 距離を求め最小スパニングツリーを計算することで, 楽曲のプレイリストを作成することができるようになった.

今回は楽曲あいだの距離を  $L_{total} / L_{mutual}$  としたが, これによってスムーズな楽曲サジェスチョンを表現できた. 例えば,  $L_{total}$  が少なくとも共通のファンが多い曲間の距離はかなり小さくなった ("ZAZEN BOYS" の2曲). しかし, アーティストが共通していても連続して再生されない場合があった ("THE ALFEE" の2曲) ので, 同じアーティストの楽曲同士を優先的に結びつける項や工夫を設けたい.

今回は, 最小スパニングツリーを用いることによって, 候補曲を複数呈示しながら再生リストを作成することができた. また新たな曲候補が増えても (ストリーミングサービスに新規楽曲が追加された場合や, ユーザーのローカル環境にある曲を追加する場合など), 擬似的なプリムのアルゴリズムを用いることで対応できる. つまり, 完成している最小スパニングツリー問題に, 新しい楽曲の全結合グラフを追加し, そこから伸ばしうる最短の辺を選べば 同じ処理を使ってスパニングツリーを, つまりはプレイリストを延長できるのだ.