

IMAGINE THE FUTURE. 開かれた 未来へ.

クラスタリングで音楽が見れる!
spectral graph theory で音楽の構成を
見よう

関 正媛
2017/12/dd

- ▶ はじめに
- ▶ クラスタリングについて
- ▶ グラフの話: 実はグラフを行列にして計算できるとか
- ▶ Spectral graph theory
- ▶ 中南米音楽について
- ▶ 具体的な実装・楽曲の構成

自己紹介

- ▶ 1988 年 韓国ソウル生まれ
- ▶ 2007 年 筑波大学理工学群数学類入学
- ▶ 2011 年 筑波大学卒業
東京工業大学大学院理工学研究科数学専攻入学
- ▶ 2016 年 博士学位取得（理学）
システム計画研究所入社
- ▶ 実は中 1 から数学が好きでした（それで博士まで行きましたが、あまりの厳しさに研究者の夢なんか諦めてます、てゆーかそれだけじゃないっすけど）
- ▶ 大学 1 年の頃から中南米音楽やってます（一緒にやる人がいなくて、細々と練習を続けています & こんな分析に走ってたりします）

音楽の構成を数学で見えるんですか？

参考にした論文は

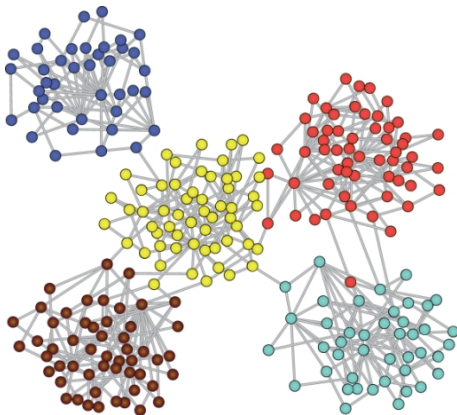
Brian McFee and Daniel P.W.Ellis 著, Analyzing song structure with spectral clustering, ISMIR 2014

です.

この論文に, クラスタリングを使って楽曲の構成 (例えば 2 : 00 ~ 2 : 20 がサビだとか) を自動抽出するアルゴリズムについて議論されています.

クラスタリングとは

クラスタリングとは、データをいくつかのクラスター（データの塊）で分割していく手法です。ラベルが付いていないデータを、距離などを用いて分割していきますので、ラベルなしデータの解析に有効です。



学習のはなし - コンピュータが学習するの?

コンピュータがある仕組みによってデータの特徴を抽出していくことを「学習」と言います。学習には、次の2つの方法があります。

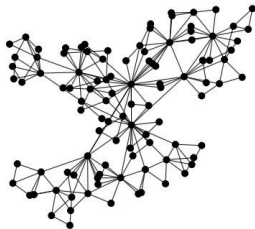
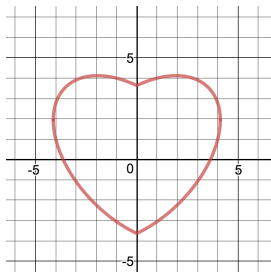
- ▶ 教師あり学習: すでにラベルが振ってあるデータからラベルごとの特性を学んでいくことです。
例) SVM, Deep Learning など
- ▶ 教師なし学習: ラベルがないデータから特性を学び、データを分類していくことです。
例) クラスタリングなど

グラフ

実は, "グラフ" と一口で言っても, グラフには 2 種類あります.

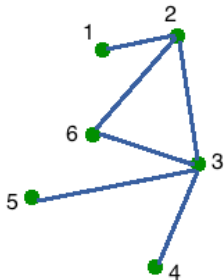
1. 関数 $f: x \mapsto y$ が与えられたとき, (x, y) の対, またはそれを可視化したものをグラフという.
(中 1 のころ "グラフは集合だ" という話を聞いてすごくショックを受けた記憶が未だに鮮明だったりする)
2. 頂点集合とエッジ (頂点と頂点をつなげるもの. 日本語では "辺" という) 集合の対.

今回話す内容は, 2 のことです.



グラフを行列にしていろんな計算ができる

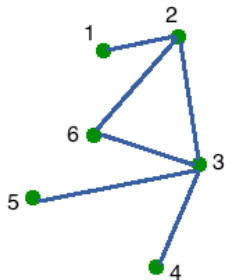
今ノードに番号をつけていますが、その番号をそれぞれ縦と横で並べ、間にエッジを挟むなら1, そうでなければ0を振っていきながら行列を作ります.



行列ならコンピュータに入力できますし、またこの行列からいろんな情報が見出せるので、便利でもあり重要でもあります.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

グラフの行列をどのように見ていくかの話をする前に、用語/記号から整理したいと思います。



- ▶ ノードの集合を V , エッジの集合を E と書きます.
- ▶ グラフを $G = (V, E)$ のように, ノードとエッジのタプルで表します.
- ▶ 1つのノードのつながっているエッジの本数を次数と言います.
各ノードの次数が対角線上に並んでいる対角行列を D と書きます.

グラフの行列とその固有値

グラフの行列の固有値は、ノードの番号のつけ方によらないので、大変重要な不変量の一つです。

例) ノード 3 つ, エッジ 2 本のグラフの場合

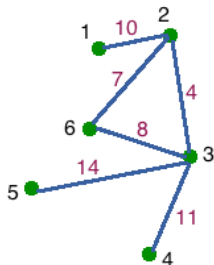
$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

この 3 つの行列の固有方程式はすべて $x^3 - 2x$ です。

グラフの行列そのものではなく、ラプラシアン $L = D - A$ や、正規化されたラプラシアン $L = I - D^{-1/2}AD^{-1/2}$ の固有値を見る場合もあります。

重み付きグラフ

”重み付きグラフ”の話をしましょう。



$$\begin{pmatrix} 0 & 10 & 0 & 0 & 0 & 0 \\ 10 & 0 & 4 & 0 & 0 & 7 \\ 0 & 4 & 0 & 11 & 14 & 8 \\ 0 & 0 & 11 & 0 & 0 & 0 \\ 0 & 0 & 14 & 0 & 0 & 0 \\ 0 & 7 & 8 & 0 & 0 & 0 \end{pmatrix}$$

重みがついていないグラフは、すべてのエッジの重みが1のグラフとして考えられます。

グラフの切断(カット)

定義

グラフ $G = (V, E)$ について, V の分割, つまり $A, B \subset V, A \cap B = \phi, A \cup B = V$ をみたす V の部分集合の対 (A, B) を切断という.

また, 集合 A, B をまたがるエッジの重みの総和を $W(A, B)$ と書く. つまり,

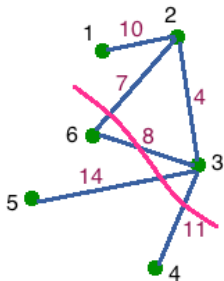
$$W(A, B) = \sum_{i \in A, j \in B} w_{ij},$$

ただし w_{ij} は i と j を結ぶエッジの重みである.

注意)

グラフの固有値とその重複度は, 頂点の番号のつけ方によらないので, この発表では頂点の集合と頂点につけた番号の集合を同一視する.

グラフの切断



例えば, 左のグラフで切断 (A, B) を

$$A = \{1, 2, 3\}, B = \{4, 5, 6\}$$

とすると,

$$W(A, B) = 7 + 8 + 14 + 11 = 40$$

となります.

3つ以上の集合で分割する時のカットの考え方

グラフの頂点を3つ以上の集合 A_1, \dots, A_k で分割するとき, 次を考える.

1. すべての $W(A_i, A_j)$ の和:

$$\text{cut}(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^k W(A_i, A_i^c).$$

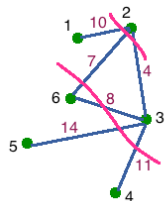
2. 次のような正規化を考えることもある.

2.1 RatioCut: 各分割の中の点の数で正規化

$$\text{RatioCut}(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^k \frac{W(A_i, A_i^c)}{|A_i|} = \sum_{i=1}^k \frac{\text{cut}(A_i, A_i^c)}{|A_i|}.$$

2.2 Ncut: 分割された点集合の次数の総和で正規化

$$\text{Ncut}(A_1, \dots, A_k) := \frac{1}{2} \sum_{i=1}^k \frac{W(A_i, A_i^c)}{\text{vol}(A_i)} = \sum_{i=1}^k \frac{\text{cut}(A_i, A_i^c)}{\text{vol}(A_i)}.$$



グラフクラスタリングのアルゴリズム

アルゴリズム (正規化なしの場合)

1. 正規化なしラプラシアン L を求める.
2. ラプラシアン L の固有値を小さい順から k 個求め, それらの固有ベクトル u_1, \dots, u_k を求める.
3. 固有ベクトル u_1, \dots, u_k を縦に並べ, $U = [u_1, \dots, u_k] = \mathbb{R}^{n \times k}$ を作る.
4. U の各行をまたベクトルとして見て, k -means を使ってクラスタリングを行う.

正規化されたラプラシアンをクラスタリングする場合は, 4 でクラスタリングをする前に各行を正規化すればよい.

ところで、なぜそのようにクラスタリングするの？

いきなりラプラシアン of 性質, 失礼します.

命題

正規化なしラプラシアン $L = D - W$ は次をみたす.

1. 任意のベクトル $f \in \mathbb{R}^n$ について,

$${}^t f L f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2$$

が成り立つ.

2. L は対称行列であり, positive semi-definite である.
3. L の最小の固有値は 0 であり, 対応する固有ベクトルは $\mathbb{1}$ である.
4. L は非負な実数固有値 $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$ をもつ.

証明. 1. L の定義より, 次が成り立つ.

$$\begin{aligned} {}^t f L f &= {}^t f D f - {}^t f W f = \sum_{i=1}^n d_i f_i^2 - \sum_{i,j=1}^n f_i f_j w_{ij} \\ &= \frac{1}{2} \left(\sum_{i=1}^n d_i f_i^2 - \sum_{i,j=1}^n f_i f_j w_{ij} + \sum_{j=1}^n d_j f_j^2 \right) = \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2. \end{aligned}$$

2. D, W ともに対象なので L も対象. positive semi-definite は 1 より.
3. すぐわかる. (D が W の各行の話を対角線上に並べたやつということからすぐわかる)
4. 上の 3 つから導かれる.

正規化されてラプラシアンについては

命題

1. 任意の $f \in \mathbb{R}^n$ について,

$${}^t f L f = \frac{1}{2} \sum_{i,j=1}^n w_{ij} \left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}} \right)^2$$

が成り立つ.

2. 0 は L の固有値であり, 付随する固有ベクトルは $D^{1/2} \mathbb{1}$ となる.
3. L は positive semi-definite であり, n 個の非負な実数固有値 $0 = \lambda_1 \leq \dots \leq \lambda_n$ をもつ.

証明は割愛します.

クラスタリングのアルゴリズム (再)

アルゴリズム (正規化なしの場合)

1. 正規化なしラプラシアン L を求める.
2. ラプラシアン L の固有値を小さい順から k 個求め, それらの固有ベクトル u_1, \dots, u_k を求める.
3. 固有ベクトル u_1, \dots, u_k を縦に並べ, $U = [u_1, \dots, u_k] = \mathbb{R}^{n \times k}$ を作る.
4. U の各行をまたベクトルとして見て, k -means を使ってクラスタリングを行う.

正規化されたラプラシアンをクラスタリングする場合は, 4 でクラスタリングをする前に各行を正規化すればよい.

アルゴリズムの解説

次の問題を考えましょう.

問題

RatioCut を最小にする分割 (A, A^c) を求めるには?

ある部分集合 $A \subset V$ について, 次のベクトル $f \in \mathbb{R}^n$ (f_1, \dots, f_n) を考えよう.

$$f_i = \begin{cases} \sqrt{|A^c|/|A|} & \text{if } v_i \in A, \\ -\sqrt{|A|/|A^c|} & \text{if } v_i \in A^c. \end{cases}$$

このとき, 正規化なしラプラシアン L について次を得る.
(次のページにつづく)

$$\begin{aligned}
{}^t f L f &= \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2 \\
&= \frac{1}{2} \sum_{i \in A, j \in A^c} w_{ij} \left(\sqrt{\frac{|A^c|}{|A|}} + \sqrt{\frac{|A|}{|A^c|}} \right)^2 + \frac{1}{2} \sum_{i \in A^c, j \in A} w_{ij} \left(-\sqrt{\frac{|A^c|}{|A|}} - \sqrt{\frac{|A|}{|A^c|}} \right)^2 \\
&= \text{cut}(A, A^c) \left(\frac{|A^c|}{|A|} + \frac{|A|}{|A^c|} + 2 \right) \\
&= \text{cut}(A, A^c) \left(\frac{|A| + |A^c|}{|A|} + \frac{|A| + |A^c|}{|A^c|} \right) \\
&= |V| \cdot \text{RatioCut}(A, A^c).
\end{aligned}$$

特に

$$\begin{aligned}
\sum_{i=1}^n f_i &= \sum_{i \in A} \sqrt{\frac{|A^c|}{|A|}} - \sum_{i \in A^c} \sqrt{\frac{|A|}{|A^c|}} = 0, \\
\|f\|^2 &= \sum_{i=1}^n f_i^2 = |A| \frac{|A^c|}{|A|} + |A^c| \frac{|A|}{|A^c|} = n
\end{aligned}$$

より, この問題は次の問題に帰着する.

$$f \perp \mathbf{1} \text{ であり, } \min_{A \subset V} {}^t f L f, \|f\| = \sqrt{n}$$

となるように f の成分を分けよう.

音楽をどうやってグラフにするの？

1. 音楽を取り込む: constant-Q(C2~C8) と最初の 13 Mel frequency cepstral coefficients(MFCC) を取り込む
(python librosa で、音楽を行列の形で取り込むことができる)
2. beat で区切って平均を算出し、取り込んだ音楽の行列を圧縮する (これも librosa でできる)
3. 2 で取り込んだ行列 $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{d \times n}$ (n が時間軸. 実装上は転置を行う必要がある) について、次のような行列を構成する.

$$R_{ij}(X) := \begin{cases} 1 & x_i, x_j \text{ は互いに } k\text{-nearest} \\ 0 & \text{otherwise} \end{cases}$$

(k については後述)

4. 行列 R の各成分の周りの対角成分を見て、0 と 1 のうち多い方を選んで R' を作る.
(行列 R をなだらかにする作業らしい)

音楽のグラフ (続き)

近いベクトルだけでなく、隣り合ったベクトルという要素も加味して、次のように”無向グラフ”を作る。

$$\Delta_{ij} := \begin{cases} 1 & |i - j| = 1 \\ 0 & \text{else} \end{cases}$$

とする。つまり、 i, j が隣り合っていたら 1, そうでなければ 0 ということ。ここで、パラメータ $\mu \in [0, 1]$ について、

$$R_{ij}^{\mu} := \mu R' + (1 - \mu) \Delta_{ij}$$

を考えよう。このパラメータを、だいたい

$$\mu \sum_j R'_{ij} \approx (1 - \mu) \sum_j \Delta_{ij}$$

となるようにしておきたい。

グラフ G の点 i の次数の和を $d_i(G)$, つまり $d_i(G) := \sum_j G_{ij}$ とする. このとき,

$$\min_{\mu \in [0,1]} \frac{1}{2} \sum_i (\mu d_i(R') - (1 - \mu) d_i(\Delta))^2$$

となる μ を求めると, μ は次のようになる.

$$\mu = \frac{\langle d(\Delta), d(R') + d(\Delta) \rangle}{\|d(R') + d(\Delta)\|^2},$$

(ただし $d(\cdot) := [d(\cdot)]_{i=1}^n$) となる.

次に, R' に対して, 次のように行列 S^{rep} を求めて R' をマスクする.

$$S_{ij}^{\text{rep}} := \exp\left(-\frac{1}{2\sigma^2}\|x_i - x_j\|^2\right),$$

ただし σ^2 は x_i ごとの k -nearest との 2 乗距離の平均であり, k は beat n について $1 + \lceil \log_2 n \rceil$ である.

また, MFCC についても, 同じ要領でガウシアンカーネル S^{loc} を求めておく.

ここまでできたら, 次のように音楽の (無向/重み付き) グラフ A を作る.

$$A_{ij} := \mu R'_{ij} S_{ij}^{\text{rep}} + (1 - \mu) \Delta_{ij} S_{ij}^{\text{loc}}.$$

中南米音楽について

- ▶ 対象としている音楽は、アンデス山脈で伝承されている音楽を継承した民族音楽のジャンルである'folklore'
- ▶ folkloreが演奏されている国は、ボリビア、ペルー、チリ、エクアドル、アルゼンチンなど数々の国
- ▶ 基本的には、ケーナやサンポーニャといった管楽器と、ギターやチャランゴなどの弦楽器、その他様々なパーカッションで構成されている
- ▶ 踊りと一緒に発展してきた総合芸術的な性格をもつ
- ▶ 地形や地域の事情によって、様々な様式の音楽が演奏されている
 - ▶ ボリビアは国民性が比較的保守的だと言われていて、楽器の構成もオーソドックスな構成を好むグループが多いよう
 - ▶ エクアドルは、欧米の影響をもっとも強く受けていて、バイオリンやマンドリンを積極的に取り入れている

中南米音楽奏者の事情いろいろ

- ▶ 中南米音楽は歌と踊りが一緒になっている総合芸術なのに、奏者の中には音楽しかできない人も多い
- ▶ 五線譜に表現できないリズムもあるが、音楽を習っている人たちはあらゆる音楽をまず五線譜から理解しようとする

→ 違う紀元を持つ違う国の音楽を、ただリズムの刻み方が同じく聞こえるという理由だけで一色単にすることも多い

- ▶ エクアドルのサンファニートとボリビアのトバスを演奏するとき特に顕著 (実際 tobas-sanjuanito も存在)
- ▶ 楽曲の構成の違いは聞いてわかる程度ではある (2, 3 年目以上の熟練者の場合)

→ サンファニートとトバスの可視化を通して、その違いをわかりやすくしてみたかった

サンファニートとトバスの違い

- ▶ サンファニートは, 4 小節程度の小さな”パーツ”(だいたい 3, 4 個程度) の繰り返しが 3 回以上繰り返されることもしばしばある
- ▶ トバスは, ”半分に分けるとだいたい同じ”場合が多い

クラスタリングの例

例をいくつかあげる.

- ▶ Purimuy (エクアドル伝統音楽/サンファニート)
- ▶ Hija del Sol (アワティーニャス (ボリビア) / サンファニート)
- ▶ Paloma Volvera (アワティーニャス (ボリビア) / トバス)

クラスタリング結果を眺めて見て



イントロ - A メロ - A' メロ - イントロ - B メロ - ...

といったパターンを自動的に検出して欲しかったけどな...

→ Music annotation そのものについてもうちょっと研究する必要があるかもしれない

- ▶ しかし、少なくとも"このパターンが頻繁に繰り返されている"などの可視化はできている感じ

これからやりたいこと

- ▶ プログラミングの基礎を固める: プログラミング言語のバックボーンとなっている数学 (カテゴリー理論など) を勉強したい
- ▶ 中南米音楽のアノテーションデータを構築し, それらを”言語”のように考えて正規表現的に分析したい (アイデアはあるが, アノテーションの基準をきちんと決めるなどといった細かいタスクが一杯残っている)