

# スペクトラル・クラスタリング

宮澤 彬

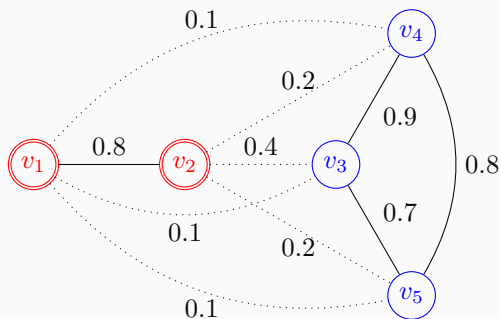
総合研究大学院大学 博士前期 1 年

`miyazawa-a@nii.ac.jp`

August 18, 2015

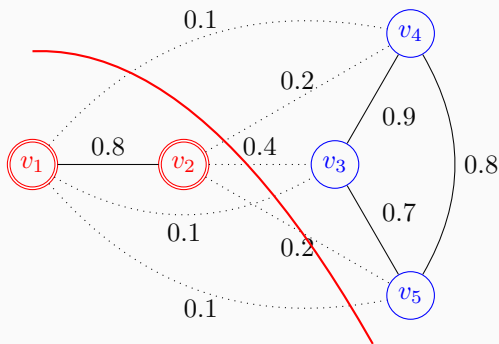
# 動機

各データを点とみなし，いくつかを辺で結ぶとグラフ理論でいうところのグラフができる．さらに辺の両端の点の類似度に応じて，重みを与えてやるとクラスタリングの問題は重みの小さい，つまり結びつきの弱い部分でグラフを切断する問題に帰着させることができる．

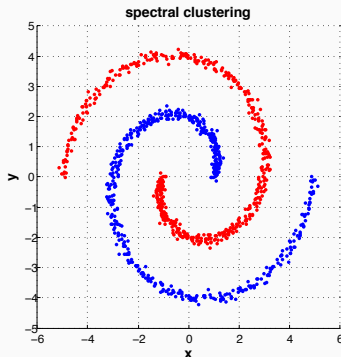
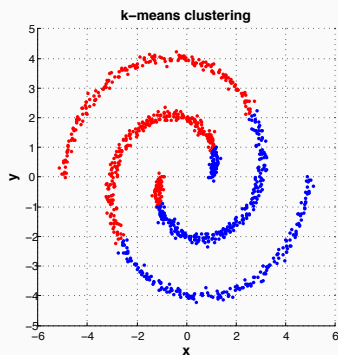


# 動機

各データを点とみなし，いくつかを辺で結ぶとグラフ理論でいうところのグラフができる．さらに辺の両端の点の類似度に応じて，重みを与えてやるとクラスタリングの問題は重みの小さい，つまり結びつきの弱い部分でグラフを切断する問題に帰着させることができる．



グラフの連結構造を活用すればクラスタリングの精度が向上する。以下の図は、 $k$  平均法ではうまくいかないが、スペクトラル・クラスタリングでは期待通りのクラスタが得られる例である。この図は Murphy (2012) から引用した。



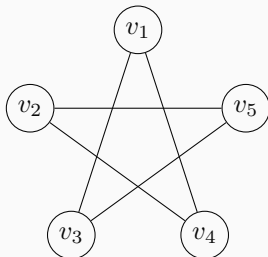
切断の問題は、グラフ理論の分野で既に多くの研究結果があり、それらを使うことができるという利点もある。

# グラフとは

グラフは、頂点と呼ばれる対象からなる有限非空集合  $V$  と、辺と呼ばれる対象からなる集合  $E$  の組  $(V, E)$  である。辺とは、 $V$  の異なる 2 つの元の非順序対である。辺  $(v_i, v_j)$  を  $e_{ij}$  のように書く。

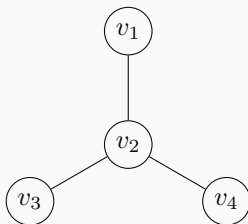
例えば以下のグラフを図示すると下の図のようになる。

$$G = (V, E), V = \{v_1, v_2, v_3, v_4, v_5\}, E = \{e_{13}, e_{14}, e_{24}, e_{25}, e_{35}\}$$



# 行列によるグラフの表現

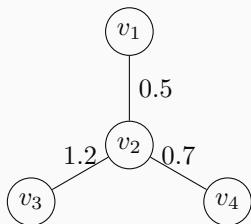
$A = (a_{ij}) = \mathbb{1}_E(e_{ij})$  とすれば、グラフの辺の構造を行列で表現することができる。この  $A$  を隣接行列 (adjacency matrix) と呼ぶ。



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

# 行列によるグラフの表現

データ  $x_1, \dots, x_n$  に頂点  $v_1, \dots, v_n$  を対応させる. 2つのデータ  $x_i, x_j$  の類似度  $s_{ij}$  によっての辺  $e_{ij}$  に重み  $w_{ij} \geq 0$  を付与する. このとき  $n$  次実対称行列  $W = (w_{ij})$  を**重み付き隣接行列** (weighted adjacency matrix) と呼ぶ.



$$W = \begin{pmatrix} 0 & 0.5 & 0 & 0 \\ 0.5 & 0 & 1.2 & 0.7 \\ 0 & 1.2 & 0 & 0 \\ 0 & 0.7 & 0 & 0 \end{pmatrix}$$

# 行列によるグラフの表現

グラフの分割を考えるにあたり、重み付き隣接行列の他に、次数行列という行列も必要になるのでここで紹介しておく。

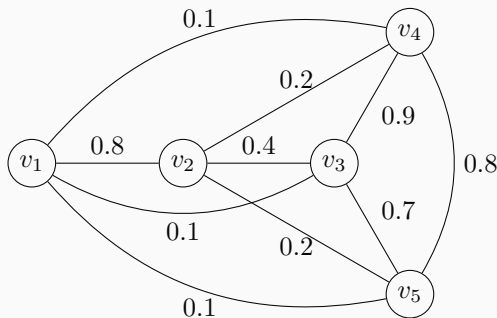
任意の頂点  $v_i \in \{v_1, \dots, v_n\}$  について、その**次数** (degree) を

$$d_i = \sum_{j=1}^n w_{ij}$$

と定める。次数を対角成分に持つ  $n$  次正方行列  $D = (d_i \delta_{ij})$  を**次数行列** (degree matrix) と呼ぶ。ただし  $\delta_{ij}$  はクロネッカーのデルタである。



# 重み付き隣接行列と次数行列



$$D = \begin{pmatrix} 1.1 & 0 & 0 & 0 & 0 \\ 0 & 1.6 & 0 & 0 & 0 \\ 0 & 0 & 2.1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1.8 \end{pmatrix}, W = \begin{pmatrix} 0 & 0.8 & 0.1 & 0.1 & 0.1 \\ 0.8 & 0 & 0.4 & 0.2 & 0.2 \\ 0.1 & 0.4 & 0 & 0.9 & 0.7 \\ 0.1 & 0.2 & 0.9 & 0 & 0.8 \\ 0.1 & 0.2 & 0.7 & 0.8 & 0 \end{pmatrix}$$

# Cut とは

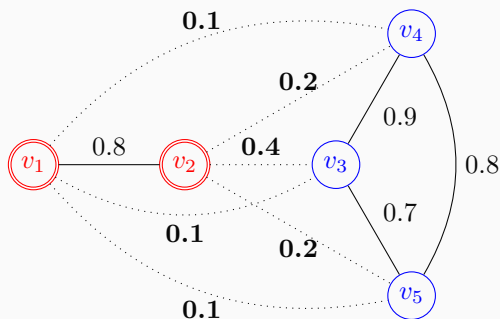
どこで切断するかを決めるため、以下で定められる量を考えよう.

$$\text{cut}(A, \bar{A}) = \sum_{\{(i,j); v_i \in A, v_j \in \bar{A}\}} w_{ij} \quad (1)$$

ここで  $\bar{A} = V \setminus A$  である. (1) はクラスター  $A$  と  $\bar{A}$  を結ぶ重みの総和であり, これが小さくなるように切断するのが望ましいと考えられる.

# Cut の例

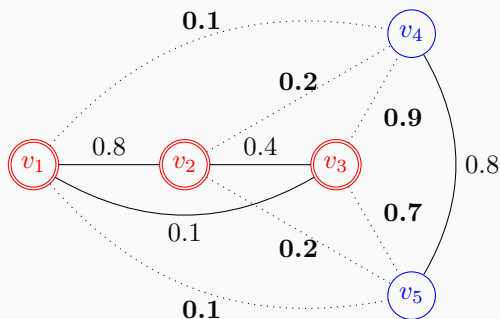
$A_1 := \{v_1, v_2\}$  とする.



$$\begin{aligned}\text{cut}(A_1, \overline{A_1}) &= w_{13} + w_{14} + w_{15} + w_{23} + w_{24} + w_{25} \\ &= 0.1 + 0.1 + 0.1 + 0.4 + 0.2 + 0.2 = 1.1\end{aligned}$$

# Cut の例

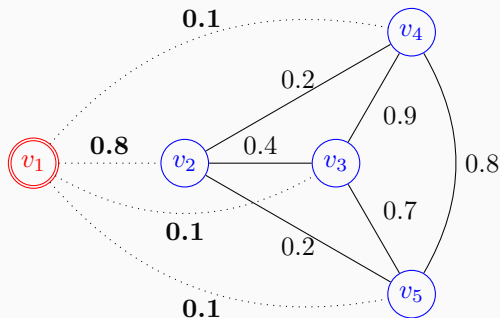
$A_2 := \{v_1, v_2, v_3\}$  とする.



$$\begin{aligned}\text{cut}(A_2, \overline{A_2}) &= w_{14} + w_{15} + w_{24} + w_{25} + w_{34} + w_{35} \\ &= 0.1 + 0.1 + 0.2 + 0.2 + 0.9 + 0.7 = 2.2 > \text{cut}(A_1, \overline{A_1})\end{aligned}$$

# Cut の例

この基準には、1つの頂点と残りの頂点全部、という分割になりやすい傾向がある。以下は  $A_3 := \{v_1\}$  とした例である。



$$\begin{aligned}\text{cut}(A_3, \bar{A}_3) &= w_{12} + w_{13} + w_{14} + w_{15} \\ &= 0.8 + 0.1 + 0.1 + 0.1 = 1.1 = \text{cut}(A_1, \bar{A}_1)\end{aligned}$$

この問題を解決するため新しく **normalized cut** という基準を導入する。これは次の値

$$\text{NCut}(A, \bar{A}) := \frac{\text{NCut}(A, \bar{A})}{\text{vol } A} + \frac{\text{NCut}(\bar{A}, A)}{\text{vol } \bar{A}}$$

が小さいほうが好ましいとする方法である。ただし

$$\text{vol } A = \sum_{\{i; v_i \in A\}} d_i = \sum_{\{i; v_i \in A\}} \sum_{\{j; v_j \in V\}} w_{ij}$$

である。上で見た例について、これを計算すると

$$\begin{aligned} \text{NCut}(A_1, \bar{A}_1) &= \frac{1.1}{1.1 + 1.6} + \frac{1.1}{2.1 + 2 + 1.8} = \frac{946}{1593} < 1, \\ \text{NCut}(A_3, \bar{A}_3) &= \frac{1.1}{1.1} + \frac{1.1}{1.6 + 2.1 + 2 + 1.8} = 1 + \frac{11}{75} > 1 \end{aligned}$$

となり、より均衡がとれている  $A_1$  が好ましいとされる。

一般に  $A_1, \dots, A_k$  について

$$\text{NCut}(A_1, \dots, A_k) = \sum_{i=1}^k \frac{\text{cut}(A_i, \overline{A_i})}{\text{vol } A_i}$$

と定める.

今までの話から, グラフの切断の問題は次の最適化問題に帰着させられる.

$$\underset{A_1, \dots, A_k}{\text{minimize}} \quad \text{NCut}(A_1, \dots, A_k) \tag{2}$$

しかし, これは NP 困難であることが知られている.

# グラフ・ラプラシアン

そこで**グラフ・ラプラシアン**というものを導入し、(2)を少し書き換えることを試みる。グラフ・ラプラシアンは

$$L := D - W$$

で与えられる。

定義からすぐに以下の性質が分かる。

- ▶  $L$  は  $n$  次実対称行列である。
- ▶  $L$  の固有値  $0$  と、それに対応する固有ベクトル  $\mathbf{1} = (1 \ \cdots \ 1)' \in \mathbb{R}^n$  をもつ。



# グラフ・ラプラシ안의性質

**補題** 任意のベクトル  $f \in \mathbb{R}$  について以下が成り立つ.

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

**証明**

$$\begin{aligned} f'Lf &= f'Df - f'Wf \\ &= \sum_{i=1}^n d_i f_i^2 - \sum_{i,j=1}^n f_i f_j w_{ij} \\ &= \left( \sum_{i=1}^n d_i f_i^2 - \sum_{i,j=1}^n f_i f_j w_{ij} + \sum_{j=1}^2 d_j f_j^2 \right) \\ &= \frac{1}{2} \sum_{i,j=1}^n w_{ij} (f_i - f_j)^2 \end{aligned}$$

# グラフ・ラプラシ안의性質

任意の  $A_\kappa \subset V$  に対して,  $h_\kappa := (h_{1,\kappa} \cdots h_{n,\kappa})' \in \mathbb{R}^n$  を以下で定める.

$$h_{i,\kappa} := \frac{\mathbb{1}_{A_\kappa}(v_i)}{\sqrt{\text{vol } A_\kappa}}$$

すると以下が成り立つ.

$$\begin{aligned} h'_\kappa L h_\kappa &= \frac{1}{2} \sum_{i,j=1}^n w_{ij} (h_{i,\kappa} - h_{j,\kappa})^2 \\ &= \frac{1}{2} \sum_{\{(i,j); v_i \in A_\kappa, v_j \in \overline{A_\kappa}\}} w_{ij} \left( \frac{1}{\sqrt{\text{vol } A_\kappa}} \right)^2 \\ &\quad + \frac{1}{2} \sum_{\{(i,j); v_i \in \overline{A_\kappa}, v_j \in A_\kappa\}} w_{ij} \left( \frac{1}{\sqrt{\text{vol } \overline{A_\kappa}}} \right)^2 \\ &= \frac{\text{cut}(A_\kappa, \overline{A_\kappa})}{\text{vol } A_\kappa}. \end{aligned}$$

# 問題の再定式化

さらに以下が成り立つ.

$$\begin{aligned} h'_\ell D h_\kappa &= \sum_{i=1}^n d_i h_{i,\ell} h_{i,\kappa} \\ &= \sum_{i=1}^n d_i \frac{\mathbb{1}_{A_\ell}(v_i) \mathbb{1}_{A_\kappa}(v_i)}{\sqrt{\text{vol } A_\ell} \sqrt{\text{vol } A_\kappa}} \\ &= \frac{1}{\sqrt{\text{vol } A_\ell} \sqrt{\text{vol } A_\kappa}} \sum_{\{i; v_i \in A_\ell \cap A_\kappa\}} d_i \\ &= \delta_{\ell\kappa}. \end{aligned}$$

よって (2) は以下のように書き換えられる.

$$\begin{aligned} &\underset{A_1, \dots, A_k \subset V}{\text{minimize}} \quad \text{tr}(H' L H) \\ &\text{subject to } H' D H = I \end{aligned} \tag{3}$$

# 問題の再定式化

しかし (3) は依然として NP 困難であるから,  $H$  が  $\mathbb{R}^{n \times k}$  を動くものとする. すなわち

$$\begin{aligned} & \underset{H \in \mathbb{R}^{n \times k}}{\text{minimize}} \quad \text{tr}(H' L H) \\ & \text{subject to} \quad H' D H = I \end{aligned} \tag{4}$$

を解くことにする.

解きやすくするため, 以下では  $T := D^{1/2} H$  と置いた

$$\begin{aligned} & \underset{T \in \mathbb{R}^{n \times k}}{\text{minimize}} \quad \text{tr}(T' D^{-1/2} L D^{-1/2} T) \\ & \text{subject to} \quad T' T = I \end{aligned} \tag{5}$$

を考える.

## 2 次形式に関する不等式

**定理** 行列  $A \in \mathbb{C}^{n \times n}$  をエルミート行列とし、その固有値を  $\lambda_1, \dots, \lambda_n$ , 対応する固有ベクトルを  $u_1, \dots, u_n$  ( $u_i u_j = \delta_{ij}$ ) とする<sup>1</sup>. このとき固有値について  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$  が成り立っているならば、任意の  $q \in \{1, \dots, n\}$  と任意の正規直交系  $\{x_1, \dots, x_q; x_i \in \mathbb{C}^n, x_i x_j = \delta_{ij}\}$  に対して

$$\sum_{i=1}^q \lambda_i \leq \sum_{i=1}^q x_i' A x_i \quad (6)$$

が成り立つ. さらに  $x_i = u_i$  ( $i = 1, \dots, q$ ) ならば, (6) において等号が成立する.

**証明** 例えば Simovici and Djeraba (2008) を参照せよ. ■

---

<sup>1</sup> エルミート行列の固有ベクトルはそれぞれ互いに直交する.

$$\lambda_i \langle u_i, u_j \rangle = \langle \lambda_i u_i, u_j \rangle = \langle A u_i, u_j \rangle = \langle u_i, A^* u_j \rangle = \langle u_i, \lambda_j u_j \rangle = \overline{\lambda_j} \langle u_i, u_j \rangle$$

エルミート行列の固有値は実数なので  $\lambda_i \neq \lambda_j$  ならば,  $\lambda_i \neq \lambda_j$  となり,  $\langle u_i, u_j \rangle = 0$ .

# 最適化問題の解

前のページの定理により (5) の解が求めることができる。つまり、行列  $D^{-1/2}LD^{-1/2}$  の固有値を小さいほうから  $\lambda_1, \dots, \lambda_n$  と並べ、それぞれに対応する正規化された固有ベクトルを  $u_1, \dots, u_n$  とするとき、 $T = (u_1 \cdots u_n)$  とすれば  $\text{tr}(T'D^{-1/2}LD^{1/2}T)$  が最小になるのである。

ここで  $y_i := D^{-1/2}u_i$  と置くと

$$D^{-1/2}LD^{-1/2}u_i = \lambda_i u_i$$

$$D^{-1}LD^{-1/2}u_i = \lambda_i D^{-1/2}u_i$$

$$D^{-1}Ly_i = \lambda_i y_i$$

$$Ly_i = \lambda_i Dy_i$$

となり、 $H = (D^{-1/2}u_1 \cdots D^{-1/2}u_n) = (y_1 \cdots y_n)$  は一般化された固有値問題  $Ly = \lambda Dy$  の解を並べたものであることが分かる。

# 類似度行列

アルゴリズムを紹介する前に、類似度行列を構成する主要な 3 つの手法を示す。

1.  $\varepsilon$ -neighbor

半径  $\varepsilon$  以内にある点のみに重みを与える。

2.  $k$ -nearest neighbor

類似度が高い  $k$  個の点のみに重みを与える。類似度行列が対称になるように注意する。例えば  $v_i$  と  $v_j$  がお互いに  $k$ -nearest neighbor に含まれる場合にのみ重さを与えるようにすればよい。

3. 完全連結

任意の 2 点間の類似度を計算する方法である。例えば

$$s_{ij} = \exp \left( -\frac{\|x_i - x_j\|^2}{2\sigma^2} \right)$$

として計算しておく。

# アルゴリズム

Normalized spectral clustering according to Shi and Malik

- 1 Construct a similarity graph.
- 2 Compute the graph Laplacian  $L$ .
- 3 Compute the first  $k$  generalized eigenvectors  $u_1, \dots, u_k$  of the generalized eigenproblem  $Lu = \lambda Du$ .
- 4 Let  $H := (h_{i\kappa}) = (u_1 \cdots u_k) \in \mathbb{R}^{n \times k}$ .
- 5 Let  $y_i := (h_{i,1} \cdots h_{i,k})' \in \mathbb{R}^k$  ( $i = 1, \dots, n$ ).
- 6 Cluster the points  $y_1, \dots, y_n$  with the  $k$ -means algorithm into clusters  $C_1, \dots, C_k$ .

時間計算量について考察する。類似度行列の作成は、完全連結で計算した場合  $O(dn^2)$  となる。固有ベクトルの計算は、手法によって違いはあるものの大体  $O(n^3)$  程度である。 $k$  平均法によるクラスタリングは繰り返しの回数を  $I$  として  $O(Ikdn)$  である。

データが増えると固有ベクトルの計算が大きな負担になってしまう問題がある。



スペクトラル・クラスタリングにはいくつかの似た変種がある。例えば normalized cut のかわりに、次の RatioCut を使う方法がある。

$$\text{RatioCut}(A_1, \dots, A_n) := \sum_{i=1}^k \frac{\text{cut}(A_i, \overline{A_i})}{|A_i|}$$

ただし  $|A_i|$  は  $A_i$  に含まれる頂点の数である。

このスライドは主に von Luxburg (2007) を参考に作った。類似の他の手法や技術的な詳細については、この資料を参考にするとよい。

- Murphy, K. P. (2012). Machine learning: a probabilistic perspective.
- Simovici, D. A. and Djeraba, C. (2008). Mathematical tools for data mining. Advanced Information and Knowledge Processing, pages 129--172.
- von Luxburg, U. (2007). A tutorial on spectral clustering. Statistics and computing, 17(4):395--416.