

## 楕円を用いたスペクトラル・クラスタリング

## Spectral Clustering by Ellipsoid

水谷友彦

Tomohiko Mizutani

東京工業大学 経営工学専攻

Department of Industrial Engineering and Management, Tokyo Institute of Technology

This manuscript explains the spectral clustering algorithm proposed in [3]. The algorithm is a variant of the normalized cut algorithm by Shi and Malik in [5] and Ng, Jordan and Weiss in [4]. We describe the algorithm with the explanation of the idea behind it. Also, we mention the features of the algorithm, and report the experimental results.

## 1. はじめに

[3]ではスペクトラル・クラスタリングのための正規化カットアルゴリズム (NC アルゴリズム) の異形種を提案した。本稿ではそのアルゴリズムについて説明する。NC アルゴリズムは Shi-Malik[5], Ng-Jordan-Weiss[4] によって提案されたクラスタリング手法である。\$m\$ 個のデータ \$\mathbf{a}\_1, \dots, \mathbf{a}\_m \in \mathbb{R}^d\$ が与えられたとき、似ているデータは同じグループに、似ていないデータは異なるグループに分類することをクラスタリングと言う。クラスタリングのためのアルゴリズムによって分類されたデータのグループをクラスターと呼ぶことにする。

スペクトラル・クラスタリングではデータ間の類似度をグラフを用いて表現する。\$m\$ 個のデータはそれぞれグラフの頂点に対応し、その辺には2つのデータの類似度を表す重みが付与されている。グラフの頂点を類似度が高いデータは同じグループに、類似度が低いデータは異なるグループとなるように分類することを考える。この作業は正規化カット関数の最小化問題として定式化することができ、最適解が \$m\$ 個のデータのいくつかのクラスターへの最適割り当てを与える。

一般に正規化カット最小化問題を厳密に解くことは困難である。したがって、この緩和問題を考える。緩和問題はデータの類似度を表すグラフに付随するグラフ・ラプラシアンと呼ばれる行列の固有値問題に帰着することができる。NC アルゴリズムでは正規化したグラフ・ラプラシアンの固有ベクトルを計算し、得られた固有ベクトルに対して K 平均を適用することでクラスターを構築する。[3]で提案した手法では、K 平均法の代わりに固有ベクトルに対する体積最小閉包楕円を計算することでクラスターを構築する。

NC アルゴリズムは K 平均法を利用する。K 平均法では初期点の選び方に最終的に得られるクラスターの様子が依存することが知られている。したがって、NC アルゴリズムではある初期点では良いクラスタリング性能を示すが、別の初期点ではそうではないということが起こりえる。一方で、提案手法では初期点の選び方にクラスタリング性能が影響を受けることはない。また、提案手法は分離可能な非負行列分解に対する楕円丸め法 [2] のある意味での一般化として見ることができる。

本稿は次のように構成されている。2節では NC アルゴリズムを復習し、提案手法を3節で述べる。4節では提案手法の特徴について言及し、5節で実験結果を報告する。

## 2. 正規化カットアルゴリズム

NC アルゴリズムについて復習する。与えられた \$m\$ 個のデータ \$\mathbf{a}\_1, \dots, \mathbf{a}\_m \in \mathbb{R}^d\$ に対して、以下のような重み付きグラフ \$G\$ を構築する。\$G\$ は \$m\$ 個の頂点 \$v\_1, \dots, v\_m\$ を持ち、それらは \$m\$ 個のデータ \$\mathbf{a}\_1, \dots, \mathbf{a}\_m\$ に各々対応している。2つの頂点 \$v\_i\$ と \$v\_j\$ を結ぶ辺には正の重み \$k\_{ij}\$ が付与されている。この重みはデータ \$\mathbf{a}\_i\$ と \$\mathbf{a}\_j\$ の類似度に対応している。2つのデータ \$\mathbf{a}\_i\$ と \$\mathbf{a}\_j\$ が良く似ている場合は重みは大きな正の値となり、そうでない場合は零に近い値となる。重みの値は2つのデータ \$\mathbf{a}\_i\$ と \$\mathbf{a}\_j\$ に対する関数 \$k\$ によって与えられ、この関数は特に類似度関数と呼ばれる。\$G\$ を構築する際、小さい重みは除外されることが多い。データ \$\mathbf{a}\_i\$ に対して、それ以外のデータ \$\mathbf{a}\_j\$ を重み \$k(\mathbf{a}\_i, \mathbf{a}\_j)\$ が高いものから順に並べて、そのうちの上位 \$p\$ 個のデータを選択し集合 \$\mathcal{N}\_p(\mathbf{a}\_i)\$ を構築する。2つの頂点 \$v\_i\$ と \$v\_j\$ の間の重み \$k\_{ij}\$ は

$$k_{ij} = \begin{cases} k(\mathbf{a}_i, \mathbf{a}_j), & \text{if } \mathbf{a}_i \in \mathcal{N}_p(\mathbf{a}_j) \text{ または } \mathbf{a}_j \in \mathcal{N}_p(\mathbf{a}_i), \\ 0, & \text{otherwise} \end{cases}$$

として定める。\$p\$ は事前に定める必要があるパラメータで、近傍点数と呼ぶことにする。

\$G\$ を上述のように構築した重み付きグラフとする。その \$m\$ 個の頂点から構成される集合 \$\mathcal{S}\$ に対して、\$r\$ 個の部分集合 \$\mathcal{S}\_1, \dots, \mathcal{S}\_r\$ を考える。この部分集合は互いに素で、その和集合は \$\mathcal{S}\$ となるとする。\$\mathcal{S}\_1, \dots, \mathcal{S}\_r\$ に対して、正規化カット関数 \$f\$ は

$$f(\mathcal{S}_1, \dots, \mathcal{S}_r) = \sum_{i=1}^r \frac{\text{cut}(\mathcal{S}_i, \mathcal{S}_i^c)}{\text{vol}(\mathcal{S}_i)}$$

として定められる。\$\text{cut}(\mathcal{S}, \mathcal{S}^c)\$ は頂点の部分集合 \$\mathcal{S}\$ とその補集合 \$\mathcal{S}^c\$ の間の重みの和を表している。

$$\text{cut}(\mathcal{S}, \mathcal{S}^c) = \frac{1}{2} \sum_{i \in \mathcal{S}, j \in \mathcal{S}^c} k_{ij}$$

ここで \$i \in \mathcal{S}\$ は \$\mathcal{S}\$ 内の頂点 \$v\_i\$ の添字 \$i\$ を意味する。\$\text{vol}(\mathcal{S})\$ は \$\mathcal{S}\$ 内の頂点の次数の和を表している。頂点 \$v\_i\$ の次数 \$d\_i\$ とは \$v\_i\$ とその他の頂点との間の重みの和である。

$$\text{vol}(\mathcal{S}) = \sum_{i \in \mathcal{S}} d_i \text{ where } d_i = \sum_{j=1}^m k_{ij}$$

正規化カット関数  $f$  の最小化は、 $S_i$  内のデータとその補集合  $S_i$  内のデータの類似度が低く、かつ、 $S_i$  の規模が大きくなるような  $S_1, \dots, S_r$  を見つける問題と解釈できる。

$m$  個のデータ  $\mathbf{a}_1, \dots, \mathbf{a}_m$  から  $r$  個のクラスター  $S_1, \dots, S_r$  を見つける問題を正規化カット関数の最小化として定式化する。以下のような要素  $h_{ij}$  からなる行列  $\mathbf{H} \in \mathbb{R}^{m \times r}$

$$h_{ij} = \begin{cases} 1/\sqrt{\text{vol}(S_j)}, & \mathbf{a}_i \in S_j, \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

を用いると、 $f$  は  $f = \text{tr}(\mathbf{H}^\top \mathbf{L} \mathbf{H})$  と書き直せる。すると、正規化カット最小化は

$$P: \text{minimize } \text{tr}(\mathbf{H}^\top \mathbf{L} \mathbf{H}) \quad \text{subject to } \mathbf{H} \text{ satisfies (1)}$$

と表せる。tr は行列のトレースを表す。L は  $G$  のグラフ・ラプラシアンで、大きさ  $m$  の対称行列である。具体的には  $\mathbf{K} = (k_{ij}) \in \mathbb{S}^m$  と対角要素が頂点の次数からなる大きさ  $m$  の対角行列  $\mathbf{D}$  に対して、 $\mathbf{L} = \mathbf{D} - \mathbf{K}$  となる。ここで  $\mathbf{D}$  は次数行列と呼ばれる。一般的には  $P$  の最適解を求めることは困難である。したがって、制約条件をより扱いやすいもの  $\mathbf{H}^\top \mathbf{D} \mathbf{H} = \mathbf{I}$  に置き換えた緩和問題を考える。I は単位行列を表す。

$$Q: \text{minimize } \text{tr}(\mathbf{H}^\top \mathbf{L} \mathbf{H}) \quad \text{subject to } \mathbf{H}^\top \mathbf{D} \mathbf{H} = \mathbf{I}$$

Q の最適値、最適解は正規化したグラフ・ラプラシアン  $\bar{\mathbf{L}} = \mathbf{D}^{-1/2} \mathbf{L} \mathbf{D}^{-1/2} \in \mathbb{S}^m$  の固有値分解から得られる。今、 $\bar{\mathbf{L}}$  の固有値を小さいものから順に  $\lambda_1, \dots, \lambda_m$  where  $\lambda_1 \leq \dots \leq \lambda_m$  とし、それに対応する固有ベクトルを  $\mathbf{v}_1, \dots, \mathbf{v}_m$  と記述する。 $\lambda = \lambda_1 + \dots + \lambda_r$ ,  $\mathbf{V}_r = (\mathbf{v}_1, \dots, \mathbf{v}_r) \in \mathbb{R}^{m \times r}$  に対して、Q の最適値は  $\lambda$ 、最適解は  $\mathbf{D}^{-1/2} \mathbf{V}_r$  となる。

緩和問題 Q の最適解  $\mathbf{D}^{-1/2} \mathbf{V}_r$  を利用して、元問題 P の最適解  $\mathbf{H}$  にできるだけ近くなるような解を構築することを目指す。ここで最適解  $\mathbf{H}$  を幾何的な観点から考察する。 $\mathbf{H}^\top$  の列ベクトルを  $\mathbf{f}_1, \dots, \mathbf{f}_m \in \mathbb{R}^r$  と書く。 $\mathbf{H}$  は (1) の形で与えられることから、 $\mathbf{f}_i$  は非零要素を1つだけ有し、その要素の添字  $j$  はデータ  $\mathbf{a}_i$  をクラスター  $S_j$  に割り当てることが正規化カット最小化という意味で最適であることを意味する。 $\mathbf{f}_1, \dots, \mathbf{f}_m$  の凸包は  $r$  次元空間上の  $(r-1)$  次元単体  $\Delta$  となる。 $\mathbf{f}_1, \dots, \mathbf{f}_m$  中には  $r$  個の異なるベクトルが存在し、それらが  $\Delta$  の  $r$  個の頂点に対応している。ここで、緩和問題 Q の最適解  $\mathbf{D}^{-1/2} \mathbf{V}_r$  が元問題 P の最適解  $\mathbf{H}$  の良い近似となっている状況を考える。 $\mathbf{V}_r^\top \mathbf{D}^{-1/2} \in \mathbb{R}^{m \times r}$  の列ベクトルを  $\mathbf{p}_1, \dots, \mathbf{p}_m \in \mathbb{R}^r$  と書く。すると、 $\mathbf{p}_1, \dots, \mathbf{p}_m$  は  $r$  次元空間上の  $(r-1)$  次元単体の  $r$  個の頂点の近くでそれぞれクラスターを形成していることが期待される。図 1 は  $r=3$  のときの期待される様子を表している。したがって、NC アルゴリズムでは  $\mathbf{p}_1, \dots, \mathbf{p}_m$  に対して K 平均法を適用することでクラスターを見つける。

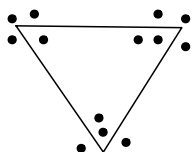


図 1:  $r=3$  のときの期待される様子。点は  $\mathbf{p}_i$  を表している。

### 3. 提案手法

提案手法は次のような考察に基づいている。 $\mathbf{p}_1, \dots, \mathbf{p}_m \in \mathbb{R}^r$  は上述の通り  $\mathbf{V}_r^\top \mathbf{D}^{-1/2}$  の列ベクトルとする。

**考察 1.**  $\mathbf{p}_1, \dots, \mathbf{p}_m \in \mathbb{R}^r$  の凸包の形状は  $(r-1)$  次元単体  $\Delta$  と似ており、また、これらのベクトルは  $\Delta$  の各頂点の近くでクラスターを形成している。

Q の最適解  $\mathbf{D}^{-1/2} \mathbf{V}_r$  が P の最適解  $\mathbf{H}$  の良い近似となっている場合、考察 1 の成立が期待できる。また、性質 1 より  $\mathbf{p}_1, \dots, \mathbf{p}_m$  は超平面上に載っているので少なくともこれらのベクトルの凸包は  $(r-1)$  次元の多面体となっている。

**性質 1** ([3] の Proposition 1).  $\mathbf{V}_r^\top \mathbf{D}^{-1/2}$  の列ベクトル  $\mathbf{p}_1, \dots, \mathbf{p}_m \in \mathbb{R}^r$  は超平面  $\{\mathbf{x} \in \mathbb{R}^r : \mathbf{e}_1^\top \mathbf{x} = \tau\}$  に載っている。ここで  $\mathbf{e}_1 = (1, 0, \dots, 0)^\top \in \mathbb{R}^r$  で、 $\tau$  は非零な実数である。

この性質はグラフ・ラプラシアン の最小固有値は零で、それに対応する固有ベクトルは全ての要素が 1 となるベクトルとなることから得られる。考察 1 が成り立つとする。すると、単体  $\Delta$  の各頂点の近くにあるデータはクラスターの代表点とみなすことができる。したがって、各データを代表点に割り当てることでクラスターを構築することができる。このような代表点は  $\mathbf{p}_1, \dots, \mathbf{p}_m$  に対する体積最小閉包楕円 (MVEE) を計算することで見つけることができる。実際、以下のような事実が [2] において示されている。

**性質 2** ([2] の Proposition 3, Corollary 4, Lemma 11).  $r$  次元空間上の  $(r-1)$  次元単体  $\Delta$  の頂点  $\mathbf{g}_1, \dots, \mathbf{g}_r$  と  $\Delta$  内の点  $\mathbf{h}_1, \dots, \mathbf{h}_\ell$  を考える。 $\mathcal{T} = \{\pm \mathbf{g}_1, \dots, \pm \mathbf{g}_r, \pm \mathbf{h}_1, \dots, \pm \mathbf{h}_\ell\}$  とする。

- $\mathcal{T}$  に対する MVEE の境界には  $\pm \mathbf{g}_1, \dots, \pm \mathbf{g}_r$  のみが載る。
- $\mathbf{h}_1, \dots, \mathbf{h}_\ell$  に多少の摂動が加わったとしても、 $\mathcal{T}$  に対する MVEE の境界には  $\pm \mathbf{g}_1, \dots, \pm \mathbf{g}_r$  のみが載る。
- $\mathbf{h}_1, \dots, \mathbf{h}_\ell$  に加わる摂動が大きいつき、 $\mathcal{T}$  に対する MVEE の境界には少なくとも  $r$  個以上の点が載る。

MVEE の計算は凸計画問題となり効率的なアルゴリズムが存在する。

提案手法では、 $\mathbf{p}_1, \dots, \mathbf{p}_m$  に対する MVEE を計算し、境界上に載っているデータを求める。このとき  $r$  個以上のデータが境界上に載っている可能性があるが、逐次射影法を利用することで適切に  $r$  個のデータを選択することが期待できる。実際、[1] は  $\mathbf{p}_1, \dots, \mathbf{p}_m$  の凸包が単体の形状に近い場合、逐次射影法はその頂点の近くにある点を見つけることができるということを示している。このようにして得られた  $r$  個のデータをクラスターの代表点とみなし、他のデータを代表点に割り当てることでクラスターを構築する。提案手法を Algorithm 1 にまとめる。

### 4. 手法の特徴

#### 4.1 クラスタリング性能の安定性

NC アルゴリズムでは Algorithm 1 のステップ 3 の後に  $\mathbf{p}_1, \dots, \mathbf{p}_m$  に対して K 平均法を適用することで  $r$  個のクラス

**Algorithm 1** 提案手法

**入力:**  $S = \{\mathbf{a}_1, \dots, \mathbf{a}_m\}$ , クラスタ数  $r$ , 近傍点数  $p$ , 類似度関数  $k$ ; **出力:**  $S_1, \dots, S_r$

- 1: 類似度関数  $k$  と近傍点数  $p$  からグラフ・ラプラシアン  $L$  と次数行列  $D$  を構築する.
- 2:  $D^{-1/2}LD^{-1/2}$  の固有値分解を計算し, 小さいものから順番に  $r$  個の固有値に対応している固有ベクトル  $\mathbf{v}_1, \dots, \mathbf{v}_r \in \mathbb{R}^m$  を求める.
- 3:  $\mathbf{V}_r = (\mathbf{v}_1, \dots, \mathbf{v}_r)$  に対して,  $\mathbf{V}_r^\top D^{-1/2}$  の列ベクトルを  $\mathbf{p}_1, \dots, \mathbf{p}_m \in \mathbb{R}^r$  とする.
- 4:  $T = \{\pm \mathbf{p}_1, \dots, \pm \mathbf{p}_m\}$  に対する MVEE を計算し, 境界上に載っている点の添字集合を  $\mathcal{I}$  とする.
- 5:  $|\mathcal{I}| = r$  ならば,  $\mathcal{J} = \mathcal{I}$  とする. そうでない場合, 逐次射影法を利用して  $\mathcal{I}$  から  $r$  個の要素を取り出し, その集合を  $\mathcal{J}$  とする.
- 6:  $\mathbf{p}_1, \dots, \mathbf{p}_m$  を代表点  $\mathbf{p}_j, j \in \mathcal{J}$  に割り当てて, クラスタ  $S_1, \dots, S_r$  を構築する.

ターを発見する. K 平均法では以下のような関数  $f$  の最小化を考える.

$$f(S_1, \dots, S_r) = \sum_{j=1}^r \sum_{\mathbf{p} \in S_j} \|\mathbf{p} - \mathbf{c}_j\|_2^2 \text{ where } \mathbf{c}_j = \frac{\sum_{\mathbf{p} \in S_j} \mathbf{p}}{|S_j|}$$

ここで,  $S_1, \dots, S_r$  は  $S = \{\mathbf{p}_1, \dots, \mathbf{p}_m\}$  における  $r$  個の互いに素な部分集合である. この関数の最小化問題を厳密に解くことは困難であることが知られている. したがって, K 平均法では  $\mathbf{c}_1, \dots, \mathbf{c}_r$  と  $S_1, \dots, S_r$  を交互に固定しながら  $f$  の最小化問題を解くことで局所的な解を求める. その際, 初期点として  $\mathbf{c}_1, \dots, \mathbf{c}_r$  を前もって選択する必要があるが, その選択の仕方がクラスタの構築に影響を与えることが知られている.

提案手法では  $\mathbf{p}_1, \dots, \mathbf{p}_m$  に対する MVEE を計算することでクラスタの構築を行う. 具体的には, 凸計画問題を解くことで MVEE を求めて, その境界上に載っているデータに対して逐次射影法を適用することでクラスタの代表点を求める. 凸計画問題の解法, 及び逐次射影法では初期点の選び方によって得られる解が影響を受けることはない. つまり K 平均法を利用する NC アルゴリズムでは初期点の選び方にクラスタリングの性能が依存するが, 提案手法ではそのようなことはない. ここで, 逐次射影法とは, 点の集合  $S = \{\mathbf{p}_1, \dots, \mathbf{p}_m\}$  が与えられたとき, その凸包の頂点に対応している点を出力するアルゴリズムである.

**4.2 分離可能な非負行列に対する楕円丸め法との関係**

**4.2.1 問題と楕円丸め法の概要**

次のような大きさが  $d \times m$  の非負行列  $A$  を考える. 非負行列とは全ての要素が非負となっている行列である.

$$A = FW \in \mathbb{R}_{+}^{d \times m} \text{ where } W = (I, K)\Pi \quad (2)$$

ここで,  $F$  は  $d \times r$  の非負行列,  $I$  は  $r \times r$  の単位行列,  $K$  は  $r \times (m - r)$  の非負行列,  $\Pi$  は  $m \times m$  の置換行列である. これは分離可能性を仮定した非負行列に対応している. 本稿では (2) の  $A$  を分離可能な行列,  $F$  を基底行列と呼ぶことにする.

分離可能な非負行列から基底行列を見つけるという問題を考える. この問題は文章データからのトピック抽出やハイパースペクトラル画像からの特徴抽出に応用を持つ. (2) の分離可能な行列  $A = FW$  は, 一般性を失うことなく  $A, F, W$  の各列ベクトルの  $\ell_1$  ノルムは 1 である (つまり各列ベクトルの要素の和が 1 である) と仮定できる.  $A$  の列ベクトルの凸包を  $\text{conv}(A)$  と書く. すると,  $\text{conv}(A)$  は  $d$  次元空間上の  $(r - 1)$  次元単体となり, その  $r$  個の頂点が  $F$  の列ベクトルに対応することが分かる. つまり,  $\text{conv}(A)$  の  $r$  個の頂点を見つけることができる基底行列  $F$  を構築することができる.

[2] では上記のような考察に基づいて楕円丸め法という手法を提案した. このアルゴリズムは性質 2 を基盤としている. つまり,  $A$  の列ベクトル  $\mathbf{a}_1, \dots, \mathbf{a}_m$  に対する MVEE を計算し, その境界上に載っている点を出力するというアイデアである. ここで  $\text{conv}(A)$  は  $d$  次元空間上の  $(r - 1)$  次元単体である. 性質 2 を適用するためには空間の次元を  $d$  から  $r$  へ削減する必要がある.  $A \in \mathbb{R}^{d \times m}$  に対して次元を削減した行列を  $Q \in \mathbb{R}^{r \times m}$  と書く. 楕円丸め法ではこの列ベクトル  $\mathbf{q}_1, \dots, \mathbf{q}_m \in \mathbb{R}^r$  に対する MVEE を計算し, 境界上に載っている  $r$  個の点を出力する. 分離可能な行列に摂動が加わった場合, 境界上には  $r$  個以上の点が載っている可能性がある. この場合, 逐次射影法を利用して  $r$  個の点を選択し出力する.

$Q$  は以下のようにして求める.  $A$  の特異値分解  $A = U\Sigma V^\top$  を計算する.  $\Sigma$  は特異値を対角要素に持つ  $d \times m$  対角行列,  $U$  は左特異ベクトルで構成される  $d \times d$  直交行列,  $V$  は右特異ベクトルで構成される  $m \times m$  直交行列である. 特異値を大きいものから順番に  $r$  個だけ選択し, それを対角要素に持つ  $r \times r$  対角行列を  $\Sigma_r$  とする. また, その特異値に対応している右特異ベクトル  $\mathbf{v}_1, \dots, \mathbf{v}_r$  を列に持つ  $m \times r$  行列を  $\mathbf{V}_r$  とする. 行列  $\Sigma_r \mathbf{V}_r^\top$  の列ベクトルが超平面上に載るようにスケーリング行列  $S \in \mathbb{R}^{m \times m}$  を施して

$$Q = \Sigma_r \mathbf{V}_r^\top S \quad (3)$$

とする.

**4.2.2 提案手法と楕円丸め法の関係**

データ  $\mathbf{a}_1, \dots, \mathbf{a}_m$  に対して Algorithm 1 を実行する際, 類似度関数を  $k(\mathbf{a}_i, \mathbf{a}_j) = \mathbf{a}_i^\top \mathbf{a}_j$ , 近傍点数  $p$  をデータの個数  $m$  と設定する. すると, ステップ 2 における正規化したグラフ・ラプラシアン  $\bar{L}$  は  $I - D^{-1/2} A^\top A D^{-1/2}$  となる. 今,  $AD^{-1/2}$  の特異値分解を  $AD^{-1/2} = U\Sigma V^\top$  と書くと,  $\bar{L}$  は

$$\bar{L} = V(I - \Sigma^\top \Sigma)V^\top$$

と書き直すことができる. したがって,  $\bar{L}$  の固有値を小さいものから順番に  $\lambda_1 \leq \dots \leq \lambda_m$  とすると

$$\lambda_i = \begin{cases} 1 - \sigma_i^2, & i = 1, \dots, d, \\ 1 & i = d + 1, \dots, m \end{cases}$$

で  $\sigma_1 \geq \dots \geq \sigma_d$  となることが分かる. このことから Algorithm 1 のステップ 2 では小さいものから順に  $\lambda_1, \dots, \lambda_r$  に対応している  $V$  の列ベクトル  $\mathbf{v}_1, \dots, \mathbf{v}_r$  を選択し  $\mathbf{V}_r$  を構築する. これは  $AD^{-1/2}$  の特異値で大きいから順に  $\sigma_1, \dots, \sigma_r$  に対応している右特異ベクトル  $\mathbf{v}_1, \dots, \mathbf{v}_r$  を選択し  $\mathbf{V}_r$  を構築することと等しい.

Algorithm 1 ではこの  $\mathbf{V}_r$  から  $\mathbf{P} = \mathbf{V}_r^\top D^{-1/2}$  を構築する. そして  $\mathbf{P}$  の列ベクトル  $\mathbf{p}_1, \dots, \mathbf{p}_m$  に対して MVEE を計算し, 境界上に載っている点を見つける. 一方, 楕円丸め法にお

いて  $A$  の列ベクトルをスケールした行列  $AD^{-1/2}$  を入力とする。  $AD^{-1/2}$  から (3) の形の行列  $Q$  を構築する際、  $S$  として  $D^{-1/2}$  を選択する。すると、性質 1 から  $Q$  の列ベクトル  $q_1, \dots, q_m$  は超平面上に載ることが分かる。楕円丸め法では  $Q$  の列ベクトル  $q_1, \dots, q_m$  に対して MVEE を計算し、境界上に載っている点を見つける。

$q_i$  は  $p_i$  に変換  $\Sigma_r$  を施したものと見ることができる。MVEE は正則変換の下で不変性が存在する。したがって、 $\Sigma_r$  が正則ならば  $p_1, \dots, p_m$  の MVEE に載っている点の集合と  $q_1, \dots, q_m$  の MVEE に載っている点の集合は一致する。つまり、Algorithm 1 のステップ 4 における集合  $\mathcal{I}$  は楕円丸め法のそれと一致する。Algorithm 1, 及び楕円丸め法の次のステップでは逐次射影法を用いて境界上の点から  $r$  個の点を選択する。このときに得られる集合は一致するとは限らない。しかし、楕円丸め法において  $Q$  を構築する際、(3) の代わりに  $Q = V_r^T S$  で  $S = D^{-1/2}$  とすると、Algorithm 1 の出力と楕円丸め法の出力は完全に一致する。この詳細は [3] の Theorem 1 と Corollary 1 に述べられている。

## 5. 実験

2 種類の実験について述べる。1 つ目の実験では考察 1 の成立と Algorithm 1 のクラスタリング性能との関係を実データを用いて検証した。この実験では MNIST データセットを用いた。これは 1 から 9 までの手書き文字の画像で構成されたデータセットである。実験では 4, 5, 6 に対応する 2832 枚の画像データを取り出し新たなデータセットを作成した。Algorithm 1 の入力パラメータは以下のように設定した。クラスター数  $r$  は 3, 近傍点数  $p$  の値は 5 あるいは 2832, 類似度関数  $k$  は  $k(a_i, a_j) = a_i^T a_j$  として設定した。データセットに対して Algorithm 1 を適用して得られた 3 つのクラスターが 4, 5, 6 に対応する画像データ集合とどのくらい似ているかを正規化相互情報量 (NMI) を用いて評価した。得られたクラスターと各画像データ集合と完全に一致する場合、NMI は 1 となる。一方で、クラスターと各画像データ集合との違いが大きい場合、NMI は 0 に近い値になる。実験では、 $p = 5$  の場合、Algorithm 1 の NMI 値は 0.934,  $p = 2832$  の場合、NMI 値は 0.460 となった。図 2 における各点は Algorithm 1 のステップ 3 で得られた  $p_1, \dots, p_{2832} \in \mathbb{R}^3$  と超平面との交点に対応している。赤, 青, 緑はそれぞれ手書き文字 4, 5, 6 に対応しているデータを表している。図中の楕円はステップ 4 で得られた 3 次元の楕円と超平面との交わりを表している。また、四角で囲まれた点はステップ 5 で選択された境界上に載っている点を表している。この図から  $p = 5$  の場合、考察 1 が成立していることが観察できる。このとき NMI 値は 0.934 であり、得られたクラスターの精度が良いことが分かる。一方で、 $p = 2832$  の場合、考察 1 が成立しているとは言い難い。実際、NMI 値も低く、得られたクラスターの精度が悪いことが分かる。

2 つ目の実験では実データを用いて Algorithm 1 と NC アルゴリズムの性能比較を行った。実験では COIL20, JAFEE, ORL, MNIST, USPS という画像から構成されるデータセットを用いた。COIL20 は 20 種類の物体に対して様々な角度から撮影して得られたデータである。JAFEE, ORL は複数の人物の顔を様々な角度から撮影して得られたデータである。MNIST, USPS は 0 から 9 までの手書き文字の画像データである。実験では各データセットに対して前もって定めた基準に基づいていくつかのクラスに分類した。COIL20 の場合は 20 種

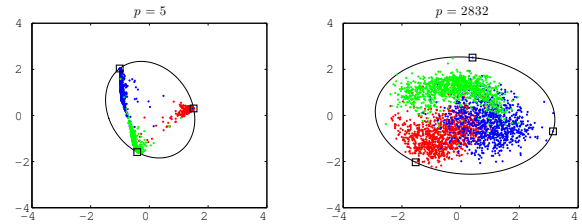


図 2: 3 種類の手書き文字画像に対する  $p_1, \dots, p_m$  と楕円の様子。左図は  $p = 5$ , 右図は  $p = 2832$  の様子を表している。

類の物体ごとにデータを分類した。JAFEE, ORL では各人物ごとにデータを分類した。MNIST, USPS では各文字ごとにデータを分類した。データセットに対してアルゴリズムが出力したクラスターと前もって作成したクラスがどれくらい似ているかを NMI を用いて評価した。Algorithm 1 の入力パラメータは以下のように設定した。クラスター数  $r$  は各データセットのクラス数, 近傍点数  $p$  の値は 5, 類似度関数  $k$  は  $k(a_i, a_j) = a_i^T a_j$  として設定した。NC アルゴリズムは K 平均法を利用する。実験では各データセットに対して異なる 100 個の初期点に対して K 平均法を実行した。表 1 は各データセットに対する Algorithm 1 と NC アルゴリズムの NMI 値をまとめている。NC アルゴリズムでは K 平均法を複数回実行したので、得られた NMI 値の平均, 最小, 最大を記載している。この表からいずれのデータセットにおいても Algorithm 1 の NMI 値は NC アルゴリズムの NMI 平均値よりも高いことが分かる。JAFEE, MNIST, USPS に対しては NC アルゴリズムの NMI 最大値は Algorithm 1 の NMI 値よりも高い。しかし、NC アルゴリズムの NMI 最小値は最大値に比べて小さい値となっており、そのため平均値が悪くなっていることが分かる。この実験結果は Algorithm 1 は NC アルゴリズムに比べて安定して高い精度のクラスタリングが実行できることを示唆している。

表 1: 提案手法と NC アルゴリズムの性能比較 (NMI)

	Algorithm 1	NC アルゴリズム		
		平均	最小	最大
COIL20	0.933	0.813	0.695	0.887
JAFEE	0.938	0.874	0.721	0.974
ORL	0.875	0.837	0.787	0.869
MNIST	0.755	0.735	0.642	0.779
USPS	0.834	0.800	0.622	0.877

## 参考文献

- [1] N. Gillis and S. A. Vavasis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):698–714, 2014.
- [2] T. Mizutani. *Journal of Machine Learning Research*, 15:1011–1039, 2014.
- [3] T. Mizutani. arXiv:1503.01531, 2015.
- [4] A. Y. Ng, M. Jordan, and Y. Weiss. In *Advances in Neural Information Processing Systems 14 (NIPS)*, pages 849–856, 2001.
- [5] J. Shi and J. Malik. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.