平成22年度茨城大学工学部情報工学科卒業研究

縮約次元数に多様性を持たせた
アンサンブルスペクトラルクラスタリング

2011年2月10日

情報工学科

著者：斉藤優 (06T4029T)

指導教員：新納浩幸准教授

平成 22 年度茨城大学工学部情報工学科卒業研究

縮約次元数に多様性を持たせたアンサンブルスペクトラルクラスタリング

著者：斉藤優 (06T4029T)
指導教員：新納浩幸准教授

論文要旨

本論文ではスペクトラルクラスタリングの次元縮約数に多様性を持たせたアンサンブルクラスタリング手法を提案する。

クラスタリングとは, 与えられたデータ (多次元データの集合であるデータセット) を, 類似性に基づいて複数のデータの集合に分割するデータ解析手法の一つである。Web 検索やデータマイニング, 文字認識や画像処理といった様々な分野で利用されており, 精度の高いクラスタリング手法が望まれている。

従来のクラスタリング手法としては, k-means が代表的である。これは計算速度も速く, 比較的良好なクラスタリング結果が得られるために, 広く用いられている。精度の高いクラスタリング手法としては, 近年, スペクトラルクラスタリングが提案されている。これはクラスタリングをグラフ分割問題として捉え, 固有値問題を解くことでクラスタリングを行う。ただしスペクトラルクラスタリングでは (分割するクラスタ数 – 1) 回分の固有値問題を解かなければならず, 計算時間の問題がある。この問題を解決したスペクトラルクラスタリング（ここではスペクトラル k-means と呼ぶ）では, ある固有値問題を解いて得られる固有ベクトルを用いて, クラスタリング対象のデータ行列を次元縮約する。その次元縮約されたデータ行列を k-means でクラスタリングを行うという手法である。精度的にはスペクトラル k-means が最も良い結果を出す傾向がある。ただしスペクトラル k-means では最適な次元縮約数が未知である。理論的には, 分割するクラスタ数に縮約するが, 現実的にはこの数に設定する必要はなく, 別の数に設定した方が最終的に得られるクラスタリングの精度が高い場合も多い。

この問題を解決するために, ここではアンサンブルクラスタリングを用いる。アンサンブルクラスタリングでは, 複数個得られているクラスタリング結果から新たなデータ行列を作成し, そのデータ行列に対してクラスタリングを行う。通常, アンサンブルされる各クラスタリング結果よりもよいクラスタリングが得られる。ここではスペクトラル k-means の次元縮約数に多様性を持たせることで, 複数個のクラスタリング結果を得る。その結果からハイパーグラフの手法により, 新たなデータ行列を作成し, クラスタリングを行う。

実験では Web で公開されている 10 個のデータセットを用いた。k-means, スペクトラルクラスタリング及びスペクトラル k-means と本手法との比較を行い, 本手法の有効性を示した。

目 次

第 1 章　序論 **1**
1.1　研究の背景と目的 1
1.2　本論文の構成 2

第 2 章　クラスタリング **3**
2.1　クラスタリング 3
2.2　クラスタリング手法の概要 4
2.3　ベクトルによるデータ表現 5
2.4　データ間の類似度 5
2.5　クラスタリング結果の評価 6

第 3 章　k-means **8**
3.1　k-means 8
3.2　kkz 13

第 4 章　スペクトラルクラスタリング **16**
4.1　スペクトラルクラスタリング 16
4.2　類似度グラフ 16
4.3　類似度行列 17
4.4　定義 17
4.5　Mcut 18
4.6　スペクトラル k-means 22

第 5 章　アンサンブルクラスタリング **25**
5.1　アンサンブルクラスタリング 25
5.2　ハイパーグラフ 25

第 6 章　実験 **27**
6.1　実験内容 27
6.2　データセット 27
6.3　実験結果 29
6.4　考察 34

第 7 章　結論 **35**

付 録 A　サンプルプログラム **38**

第1章　序論

1.1　研究の背景と目的

クラスタリングとは, 与えられたデータ (多次元データの集合であるデータセット) を, 類似性に基づいて複数のデータの集合に分割するデータ解析手法の一つである。Web 検索やデータマイニング, 文字認識や画像処理といった様々な分野で利用されており, 精度の高いクラスタリング手法が望まれている。

従来のクラスタリング手法としては, k-means[1] が代表的である。これは計算速度も速く, 比較的良好なクラスタリング結果が得られるために, 広く用いられている。精度の高いクラスタリング手法としては, 近年, スペクトラルクラスタリングが提案されている。これはクラスタリングをグラフ分割問題として捉え, 固有値問題を解くことでクラスタリングを行う。ただしスペクトラルクラスタリングでは (分割するクラスタ数 − 1) 回分の固有値問題を解かなければならず, 計算時間の問題がある。この問題を解決したスペクトラルクラスタリング（ここではスペクトラル k-means と呼ぶ）では, ある固有値問題を解いて得られる固有ベクトルを用いて, クラスタリング対象のデータ行列を次元縮約する。その次元縮約されたデータ行列を k-means でクラスタリングを行うという手法 [3, 4] である。精度的にはスペクトラル k-means が最も良い結果を出す傾向がある。

ただしスペクトラル k-means では最適な次元縮約数が未知である。理論的には, 分割するクラスタ数に縮約するが, 現実的にはこの数に設定する必要はなく, 別の数に設定した方が最終的に得られるクラスタリングの精度が高い場合も多い。

この問題を解決するために, ここではアンサンブルクラスタリング [2] を用いる。アンサンブルクラスタリングでは, 複数個得られているクラスタリング結果から新たなデータ行列を作成し, そのデータ行列に対してクラスタリングを行う。通常, アンサンブルされる各クラスタリング結果よりもよいクラスタリングが得られる。ここではスペクトラル k-means の次元縮約数に多様性を持たせることで, 複数個のクラスタリング結果を得る。その結果からハイパーグラフの手法により, 新たなデータ行列を作成し, クラスタリングを行う。

実験では Web で公開されている 10 個のデータセットを用いた。k-means, スペクトラルクラスタリング及びスペクトラル k-means と本手法との比較を行い, 本手法の有効性や問題点を調べた。実験の結果, 本手法が従来の手法より精度の高いクラスタリングを行えることが確認できた。

1.2 本論文の構成

第 2 章ではクラスタリングの基礎的なことについて説明する。第 3 章は非階層的手法の代表的なクラスタリングアルゴリズムである k-means について説明する。 第 4 章はクラスタリングをグラフ分割として考えるスペクトラルクラスタリングの手法を紹介する。第 5 章はアンサンブルクラスタリングの手法を記す。第 6 章には今まで紹介した手法とそれらを組み合わせた手法とでの比較実験について述べる。最後に第 7 章で結論を記す。

第2章　クラスタリング

2.1　クラスタリング

クラスタリングとは入力データを似ているもの同士の集合となるように分割することである。分割された部分集合はクラスタと呼ばれる。図 2.1 はデータ集合を 3 つのクラスタに分割した場合の図である。(a) は形の類似性を観点に, (b) は番号の類似性を観点においてクラスタリングを行っており, 2 パターンのクラスタリング結果を示している。

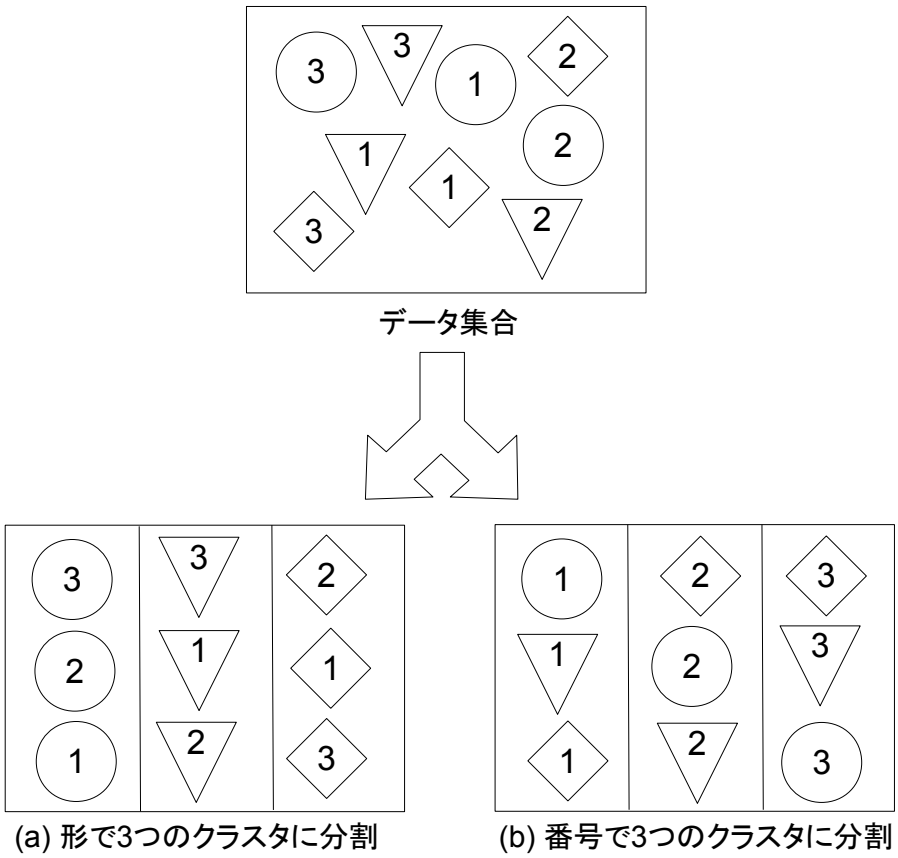

図 2.1: クラスタ数 3 の場合の分割

2.2 クラスタリング手法の概要

クラスタリングは以下の二つの観点 [1] から分類することができる。

- ソフトクラスタリングとハードクラスタリング
  - ソフトクラスタリング
    それぞれのデータが複数のクラスタに属する場合の手法。Fuzzy C-means や, EM アルゴリズムを使用した混合分布モデルによるクラスタリングなどがある。
  - ハードクラスタリング
    それぞれのデータが一つのクラスタのみに属する場合の手法。k-means やスペクトラルクラスタリングは, これに分類される。一般的なクラスタリングはハードクラスタリングを指す。
- 階層的クラスタリングと非階層的クラスタリング
  - 階層的クラスタリング
    個々のデータそれぞれを 1 つのデータのみからなるクラスタとし, クラスター間の距離 (非類似度) 関数にもとづき, 最も距離の近い 2 つのクラスターを逐次的に併合する。そして，この併合をすべての対象が 1 つのクラスターに併合されるまで繰り返すことで階層構造を得る。単連結法, 群平均法, ウォード法などがある。
  - 非階層的クラスタリング
    データ全体を 1 つのクラスタとし, それを似ているもの同士のデータが同じクラスタとなるように分割していき, 目標のクラスタ数を得る手法。分割の良さの評価関数を定め, その評価関数を最適にする分割を探索する。可能な分割の総数はデータ数に対して指数的に増加するので実際には準最適解を求めることになる。k-means やスペクトラルクラスタリング, Fuzzy C-means などが挙げられる。

2.3 ベクトルによるデータ表現

一般的にデータは多次元ベクトルで表現される。各次元はデータの特徴を示す属性である。例えば, 次のような健康診断データがあるとする。

年齢 [才]	身長 [cm]	体重 [kg]	腹囲 [cm]	血圧-1[mmHg]	血圧-2[mmHg]
43	167.7	62.1	82.3	131	91

その数値を並べた 6 次元のベクトル

$$x = (43,\ 167.7,\ 62.1,\ 82.3,\ 131,\ 91)$$

がクラスタリングで扱うデータである。このデータは 6 次元空間に存在する点として捉えることができる。

2.4 データ間の類似度

各データがどれだけ似ているかを表す数値が類似度である。ベクトル間のユークリッド距離から類似度を求める方法が一番単純であるが, ここではベクトル間の余弦尺度 [1] を使う。ベクトル a とベクトル b のなす角を θ とすると $\cos\theta$ は以下のように表すことができる。

$$\cos\theta = \frac{(a,b)}{||a||||b||}$$

a と b が正規化されていれば, 上式の分母は 1 となるので, $\cos\theta = (a,b)$ となる。また, 以下の式が成り立つ。

$$||a-b||^2 = ||a||^2 - 2(a,b) + ||b||^2 = 2(1-\cos\theta)$$

上の式から以下の式が成り立つ。

$$\cos\theta = 1 - \frac{||a-b||^2}{2}$$

$||a-b||$ は a と b 間の距離である。よって a と b が同一のデータであれば a と b 間の距離は 0 となるので $\cos\theta$ は 1 である。また, a と b が似ていないということは, それだけ a と b 間の距離が離れているということであり, $\cos\theta$ は 1 よりも小さくなっていく。このことから, $\cos\theta$ は a と b の類似度となっていることがわかる。

このように入力データが正規化されていれば, $\cos\theta = (a,b)$ なので, 類似度を内積で求めることができる。

2.5 クラスタリング結果の評価

クラスタリング結果の評価の指標として, エントロピーと純度 [1] がある。これは正解集合が存在し, データのクラスタ数が既知である場合に利用できる。

- クロス表
  エントロピーと純度を算出するためにはクロス表が必要となる。クラスタリングの結果のクラスタ集合 $C = \{C_1, \cdots, C_k\}$, 正解のクラスタ集合 $A = \{A_1, \cdots, A_k\}$, そして C_i と A_j の両方のクラスタに属するデータの個数を $X_{ij} = |C_i \cap A_j|$ とすると, クロス表は以下のようになる。

	A_1	$\cdots$	A_j	$\cdots$	A_k
C_1	X_{11}	$\cdots$	X_{1j}	$\cdots$	X_{1k}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
C_i	X_{i1}	$\cdots$	X_{ij}	$\cdots$	X_{ik}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
C_k	X_{k1}	$\cdots$	X_{kj}	$\cdots$	X_{kk}

- エントロピー
  クラスタリング結果と正解のクラスタのデータとの違いのバラツキの度合い
  各クラスタ C_i に対するエントロピー E_i

$$E_i = -\sum_{h=1}^{k} P(A_h|C_i) log P(A_h|C_i)$$

  各クラスタの重み付き平均によって全体のエントロピーを定義 (N はデータ数)

$$\sum_{i=1}^{k} \frac{|C_i|}{N} E_i = \sum_{i=1}^{k} \frac{\sum_{j=1}^{k} X_{ij}}{N} E_i \tag{2.1}$$

  (2.1) 式は $0 \sim 1$ の値をとり, 値が低いほうがクラスタリング結果がよい

- 純度
  ある正解のクラスタのデータをどの程度含むかという指標
  各クラスタ C_i に対する純度 P_i

$$P_i = \frac{1}{|C_i|} \max_h |C_i \cap A_h|$$

  各クラスタの重み付き平均によって全体の純度を定義

$$\sum_{i=1}^{k} \frac{|C_i|}{N} P_i = \frac{1}{N} \sum_{i=1}^{k} \max_h |C_i \cap A_h| \tag{2.2}$$

  (2.2) 式は $0 \sim 1$ の値をとり, 値が高いほうがクラスタリング結果がよい

以下に例を示す。

6 つのデータ $x_i(i = 1, \cdots 6)$ を 3 つのクラスタに分割したとする。クラスタリング結果は

$$C_1 = \{x_1, x_2, x_5\},\ \ C_2 = \{x_3, x_4\},\ \ C_3 = \{x_6\}$$

となった。また正解のクラスタ集合は以下の通りである。

$$A_1 = \{x_1, x_2, x_3\},\ \ A_2 = \{x_4, x_5\},\ \ A_3 = \{x_6\}$$

よってクロス表は以下のようになる。

	A_1	A_2	A_3
C_1	2	1	0
C_2	1	1	0
C_3	0	0	1

(2.1) 式からエントロピーは 0.500, (2.2) 式から純度は 0.667 となる。

第3章　k-means

3.1　k-means

非階層的クラスタリングの代表的な手法として挙げられるのが k-means[1] である。k-means は各クラスタの代表点を決定し, それぞれのデータが最も近い代表点をそのデータの初期クラスタとする。これらを初期分割として, ある評価基準に基づいてデータを分割し直していくことで, より良い分割結果を得る手法である。

k-means の評価関数は以下の式となる。$C_i(i = 1 \cdots k)$ はクラスタで $c_i(i = 1 \cdots k)$ はクラスタの代表点である。この式を最小化するようにクラスタリングを行えばよいのだが, これは NP 困難な問題である。よって, 局所解を求めることになる。

$$\sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|$$

k-means のアルゴリズムを以下に示す。

k-means

データ数を n, 目標のクラスタ数を k とする

1. 各クラスタの代表点 $c_j(j = 1 \cdots k)$ を各データ $x_i(i = 1 \cdots n)$ からランダムに選択する
2. 各 x_i と各 c_j との距離を求め, 最も距離が近い代表点を持つクラスタを x_i のクラスタとする
3. x_i へのクラスタの割り当てが変化しないなら終了し, そうでなければ c_j を各クラスタの重心に更新してステップ 2 へ

以下に例を示す。

7 個の 3 次元データ $x_i(i = 1 \cdots 7)$ を k-means で 3 つのクラスタに分割する。各データの集合を行列 X で表すと以下の通りである。

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \begin{pmatrix} 5 & 5 & 5 \\ 9 & 2 & 3 \\ 2 & 2 & 3 \\ 4 & 8 & 9 \\ 8 & 7 & 8 \\ 2 & 9 & 4 \\ 4 & 4 & 4 \end{pmatrix}$$

まずは初期値となるクラスタ $C_1,\ C_2,\ C_3$ の代表点をランダムに選択する。代表点は $c_1 = x_2$, $c_2 = x_6$, $c_3 = x_7$ とした。各代表点と各データの距離は以下のようになる。

	c_1	c_2	c_3
x_1	5.39	5.10	1.73
x_2	0	9.95	5.48
x_3	7.00	7.07	3.00
x_4	9.85	5.48	6.40
x_5	7.14	7.48	6.40
x_6	9.95	0	5.39
x_7	5.48	5.39	0

各データは一番近い代表点を持つクラスタに割り当てされるので, 初期分割は以下のようになる。

$$C_1 = \{x_2, \},\ \ C_2 = \{x_4, x_6\},\ \ C_3 = \{x_1, x_3, x_5, x_7\}$$

次に代表点を更新する。代表点はクラスタの重心に更新されるので

$$\begin{aligned} c_1 &= x_2 = (9, 2, 3) \\ c_2 &= \left(\frac{4+2}{2}, \frac{8+9}{2}, \frac{9+4}{2}\right) = (3, 8.5, 6.5) \\ c_3 &= \left(\frac{5+2+8+4}{4}, \frac{5+2+7+4}{4}, \frac{5+3+8+4}{4}\right) = (4.75, 4.5, 5) \end{aligned}$$

となる。

先ほどと同じように各代表点と各データの距離を求める。

	c_1	c_2	c_3
x_1	5.39	4.30	0.56
x_2	0	9.51	5.32
x_3	7.00	7.45	4.22
x_4	9.85	2.74	5.37
x_5	7.14	5.43	5.08
x_6	9.95	2.74	5.37
x_7	5.48	5.24	1.35

よって 2 回目のクラスタの割当は以下のようになる。

$$C_1 = \{x_2, \},\ \ C_2 = \{x_4, x_6\},\ \ C_3 = \{x_1, x_3, x_5, x_7\}$$

クラスタの割当が変わらないので, ここで終了する。

k-means は反復回数に上限を設定する場合がある。その上限を超えたならば処理を終了し, その時点までのクラスタ割当を最終的なクラスタリング結果とする。

k-means のメリットは, 特定のクラスタ数における各クラスタサイズを最小化することができ, かつクラスタの境界を明確に決定することができるという点である。計算が単純で簡単であることも利点である。デメリットは, 初期値である代表点の選択はランダムであるので, その選択によりクラスタリング結果が変わってしまう点である。また, 対象データが線形分離不可能な場合はクラスタリングはうまくいかない。つまり多様なクラスタ形状を扱えないのである。

ここで, k-means で初期値である代表点が違う場合, クラスタリング結果が異なる例を以下に示す。

前の例と同じデータを 3 つのクラスタに分割する。今度は初期代表点を $c_1 = x_1$, $c_2 = x_3$, $c_3 = x_7$ とした。各代表点と各データの距離は以下のようになる。

	c_1	c_2	c_3
x_1	0	4.69	1.73
x_2	5.39	7.00	5.48
x_3	4.69	0	3.00
x_4	5.10	8.72	6.40
x_5	4.69	9.27	6.40
x_6	5.10	7.07	5.39
x_7	1.73	3.00	0

各データは一番近い代表点を持つクラスタに割り当てされるので, 1 回目のクラスタの割り当ては以下のようになる。

$$C_1 = \{x_1, x_2, x_4, x_5, x_6\},\ \ C_2 = \{x_3\},\ \ C_3 = \{x_7\}$$

次に代表点を更新する。代表点はクラスタの重心に更新されるので

$$\begin{aligned} c_1 &= \left(\frac{5+9+4+8+2}{5}, \frac{5+2+8+7+9}{5}, \frac{5+3+9+8+4}{5}\right) = (5.6, 6.2, 5.8) \\ c_2 &= x_3 = (2, 2, 3) \\ c_3 &= x_7 = (4, 4, 4) \end{aligned}$$

となる。先ほどと同じように各代表点と各データの距離を求める。

	c_1	c_2	c_3
x_1	1.56	4.69	1.73
x_2	6.09	7.00	5.48
x_3	6.20	0	3.00
x_4	4.00	8.72	6.40
x_5	3.35	9.27	6.40
x_6	4.90	7.07	5.39
x_7	3.26	3.00	0

よって 2 回目のクラスタの割当は以下のようになる。

$$C_1 = \{x_1, x_4, x_5, x_6\},\ \ C_2 = \{x_3\},\ \ C_3 = \{x_2, x_7\}$$

ここで 2 回目のクラスタの割り当ては 1 回目とは違う割り当てとなっているので, 続行する。
代表点を更新する。

$$\begin{aligned} c_1 &= \left(\frac{5+4+8+2}{4}, \frac{5+8+7+9}{4}, \frac{5+9+8+4}{4}\right) = (4.75, 7.25, 6.5) \\ c_2 &= x_3 = (2, 2, 3) \\ c_3 &= \left(\frac{9+4}{2}, \frac{2+4}{2}, \frac{3+4}{2}\right) = (6.5, 3, 3.5) \end{aligned}$$

先ほどと同じように各代表点と各データの距離を求める。

	c_1	c_2	c_3
x_1	2.71	4.69	2.91
x_2	7.61	7.00	2.73
x_3	6.88	0	4.64
x_4	2.72	8.72	7.84
x_5	3.59	9.27	6.20
x_6	4.11	7.07	7.52
x_7	4.17	3.00	2.74

よって 3 回目のクラスタの割当は以下のようになる。

$$C_1 = \{x_1, x_4, x_5, x_6\},\ \ C_2 = \{x_3\},\ \ C_3 = \{x_2, x_7\}$$

クラスタの割当が変わらないので, ここで終了する。

以前の例ではクラスタリング結果は

$$C_1 = \{x_2, \},\ \ C_2 = \{x_4, x_6\},\ \ C_3 = \{x_1, x_3, x_5, x_7\}$$

であったが, 今回のクラスタリング結果は

$$C_1 = \{x_1, x_4, x_5, x_6\},\ \ C_2 = \{x_3\},\ \ C_3 = \{x_2, x_7\}$$

という結果となった。このことから, k-means は初期値によりクラスタリング結果が変わるということがわかる。

3.2 kkz

k-means はクラスタの初期代表点の選択で結果が変わるので, 1 回の結果で最良のものが得られるとは限らない。そこで kkz という手法 [5] を用いて, 初期代表点となるデータをあらかじめ決めておいてから k-means を実行することで, 1 回の実行で優良な結果が得られる。

kkz はどれだけデータが互いに離れているかに着眼して代表点を決定する。このようにして決まった代表点はお互いに距離が離れており, 偏りがなく散らばっているので, 同じクラスタになりにくい。

kkz のアルゴリズムを以下に示す。

kkz

1. 各データ x_i のノルムを計算し, 最大となるデータを 1 番目のクラスタの代表点 c_1 とする
2. 各 x_i と既に決まっている代表点 $c_1, \cdots, c_k$ との距離 d_{ij} を計算する
3. 各 x_i の最小の d_{ij} を選択し, d_i とする
4. d_i の中で最大となるデータを k 番目のクラスタの代表点 c_k に設定する
5. 目標のクラスタ数だけ代表点が設定されたならば終了し, そうでなければステップ 2 へ

以下に例を示す。

9 個の 2 次元データ $x_i(i = 1 \cdots 9)$ から kkz で 3 つのクラスタの代表点 c_1, c_2, c_3 を決める。データ行列 X は以下の通りである。

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{pmatrix} = \begin{pmatrix} 1 & 2 \\ 3 & 1 \\ 9 & 2 \\ 3 & 8 \\ 10 & 9 \\ 5 & 5 \\ 6 & 9 \\ 2 & 10 \\ 8 & 6 \end{pmatrix}$$

各データのノルムで最大となるデータは x_5 である。よって $c_1 = x_5$ となり, これが一つ目の代表点となる。

次に各データ x_i と代表点 c_1 との距離 d_{ij} を計算する。d_{ij} を要素とする行列を D とすると D は

$$D = \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{array} \begin{pmatrix} 11.40 \\ 10.63 \\ 7.07 \\ 7.07 \\ 0 \\ 6.40 \\ 4.00 \\ 8.06 \\ 3.61 \end{pmatrix}$$

となる。各 x_i の最小の d_{ij} を選択する。c_1 しか決まっていないで d_{i1} が選ばれる。よって d_i は

$$d_i = (11.40, 10.63, 7.07, 7.07, 0, 6.40, 4.00, 8.06, 3.61)$$

となる。d_1 が最大となるので, $c_2 = x_1$ となり, 二つ目の代表点が決定した。今度は各データ x_i と代表点 c_1, c_2 の距離を計算する。D は以下のようになる。

$$D = \begin{array}{c} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{array} \begin{pmatrix} 11.40 & 0 \\ 10.63 & 2.27 \\ 7.07 & 8.00 \\ 7.07 & 6.32 \\ 0 & 11.40 \\ 6.40 & 5.00 \\ 4.00 & 8.60 \\ 8.06 & 8.06 \\ 3.61 & 8.06 \end{pmatrix}$$

最小の d_{ij} を選択すると d_i は

$$d_i = (0, 2.27, 7.07, 6.32, 0, 5.00, 4.00, 8.06, 3.61)$$

となる。d_8 が最大となるので, $c_3 = x_8$ となり, 三つ目の代表点が決定した。すなわち代表点は

$$c_1 = x_5, \quad c_2 = x_1, \quad c_3 = x_8$$

となる。

図 3.1 はデータ X と代表点をグラフで表した図である。丸で囲まれたデータが代表点を表している。代表点同士は互いに離れており, 偏りがないことがわかる。

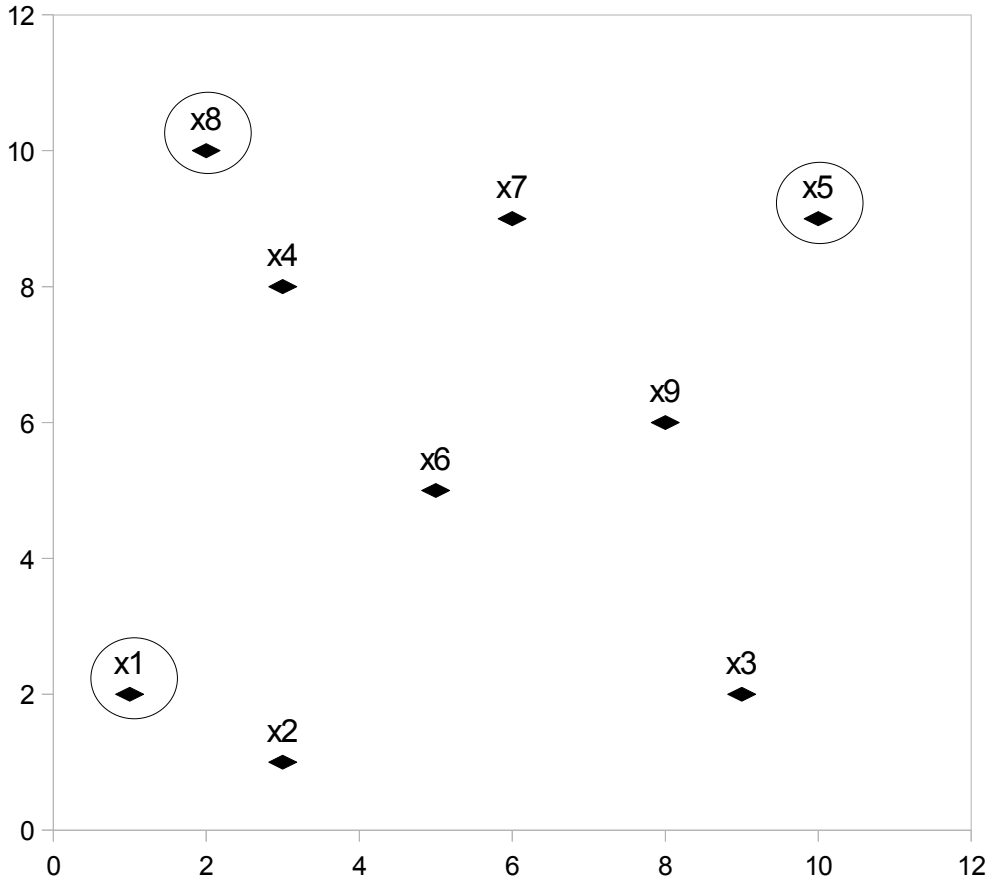

図 3.1: kkz による代表点の選択

第4章　スペクトラルクラスタリング

4.1　スペクトラルクラスタリング

各データをノードとするグラフの分割問題として考えて, クラスタリングを行う手法がスペクトラルクラスタリング [1] である。分割後のサブグラフがクラスタに相当する。実際には整数計画問題を緩和した固有値問題を解くことによって得られる固有ベクトルを使ってクラスタリングを行う。

4.2　類似度グラフ

データセット $X = \{x_1, \cdots, x_n\}$ が与えられたとき, 各データ間の関係性が保たれるようなグラフを構成することが重要である。グラフの構成方法 [3] を 3 つ紹介する。

- ε 近傍グラフ
  データ間の距離がパラメータ ε 以下であれば辺でつなぐ。このようにすることで, すべてのつながれたデータ間は大体同じくらい尺度 (最大で ε) となる。

- k 最近傍グラフ
  この方法は以下の 2 つの方法に細分化される。

  - k 最近傍グラフ
    x_j が x_i に k 番目以下に距離が近い場合または x_i が x_j に k 番目以下に距離が近い場合に x_i と x_j を辺でつなぐ。
  - 相互 k 最近傍グラフ
    x_j が x_i に k 番目以下に距離が近い場合かつ x_i が x_j に k 番目以下に距離が近い場合に x_i と x_j を辺でつなぐ。

- 完全連結グラフ
  単純にすべてのデータ間を辺でつなぐグラフ構成である。

これらのグラフから類似度行列を作ることになる。ε 近傍グラフと k 最近傍グラフは辺でつながれていないデータがあるので後々計算的には有利である。ここでは類似度行列の作成が簡単なため完全連結グラフを使うことにする。

4.3 類似度行列

n をデータ数とすると各データ $x_i(i = 1, \cdots, n)$ を点とするグラフをサブグラフに分割するので, グラフを構成する辺に重みを加えておく必要がある。そのため類似度行列 $W = w_{ij}(i, j = 1, \cdots, n)$ が必要である。w_{ij} は x_i と x_j の類似度を表している。もし $w_{ij} = 0$ であれば x_i と x_j の間に辺は存在しない。よって, x_i と x_j は違うクラスタに属すことになる。ここでは 2.4 節で紹介した余弦尺度を用いて類似度を計算している。

余弦尺度を使えば, データ X とその転置行列 $t(X)$ の積で X の類似度行列は簡単に求まる。ただし, X は行の大きさを 1 に正規化した行列である必要がある。

4.4 定義

クラスタ A と B の類似度の総和を以下の式で定義する。

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

クラスタ内の類似度の総和は以下のように定義する。

$$W(A) = cut(A, A)$$

また i 番目のデータとその他のデータ (i 番目のデータも含む) との類似度の和を $d_i = \sum_{j=1}^{n} w_{ij}$ で表すと, 次数行列 D は $d_i(i = 1, \cdots, n)$ を要素とした対角行列である。つまり D は以下のようになる。

$$D = \begin{pmatrix} d_1 & 0 & \cdots & \cdots & 0 \\ 0 & d_2 & 0 & \cdots & 0 \\ \cdots & \cdots & \ddots & \cdots & \cdots \\ \cdots & \cdots & 0 & d_{n-1} & 0 \\ 0 & \cdots & \cdots & 0 & d_n \end{pmatrix}$$

4.5 Mcut

Mcut[1] はスペクトラルクラスタリングの手法の一つで, 以下の評価関数が最小化するように, データをクラスタ A と B に分割する。

$$Mcut = \frac{cut(A,B)}{W(A)} + \frac{cut(A,B)}{W(B)} \tag{4.1}$$

つまり Mcut はクラスタ内の類似度の総和を大きくし, クラスタ間の類似度の総和を小さくすることによって, クラスタ内を密にしてクラスタ間が離れるように分割を行う。

(4.1) 式は以下の式に帰着できる

$$J_m = \frac{q^t(D-W)q}{q^tWq} \tag{4.2}$$

このとき q は n 次元のベクトルであり, q の i 番目の要素はデータ x_i がクラスタ A に含まれるなら $a = \sqrt{\frac{d_B}{d_A d}}$ となり, クラスタ B に含まれるなら $b = -\sqrt{\frac{d_A}{d_B d}}$ の離散的な値をとる。$d_X = \sum_{i \in X} d_i$、$d = d_A + d_B$ である。

つまりベクトル q はクラスタリング結果を表す指標ベクトルである。しかしこのままでは q を直接求めることはできない。よって q の要素を連続値に置き換えて近似を行う。

(4.2) 式は以下のように変形できる。

$$\frac{(W^{\frac{1}{2}}q)^t(W^{-\frac{1}{2}}DW^{-\frac{1}{2}} - I)(W^{\frac{1}{2}}q)}{(W^{\frac{1}{2}}q)^t(W^{\frac{1}{2}}q)} \tag{4.3}$$

ここでレイリー商を紹介する。$A \in R^{n \times n}$, $x \in R^n$, $x \neq 0$ とすると

$$R(x) := \frac{x^TAx}{x^Tx}$$

をレイリー商と呼び, A の最小固有値を λ_{min}, 対応する固有ベクトルを x_1 とすると, R(x) の最小値は $\lambda_{min} = R(x_1)$ で得られるという性質も持っている。

$L = W^{-\frac{1}{2}}DW^{-\frac{1}{2}} - I$, $z = W^{\frac{1}{2}}q$ とすると, (4.3) 式は $R(x) = \frac{z^TLz}{z^Tz}$ で表すことができ, この形はレイリー商であることがわかる。つまり L の最小固有値に対応する固有ベクトルが (4.3) 式の最小値を与えることになる。このベクトルをクラスタリング結果として利用できそうだが, L の最小固有値は 0 であり, 対応する固有ベクトルは定ベクトルであるのでクラスタリングには利用できない。よって, 2 番目に小さい固有値に対応する固有ベクトルをクラスタリング結果の近似値としてクラスタリングに利用する。このベクトルを Fielder ベクトルと呼ぶ。

$W^{-\frac{1}{2}}DW^{-\frac{1}{2}} - I$ の固有値問題は $I - W^{-\frac{1}{2}}DW^{-\frac{1}{2}}$ の固有値問題に置き換えることで計算を緩和できる。$I - W^{-\frac{1}{2}}DW^{-\frac{1}{2}}$ を正規化ラプラス行列と呼ぶ。

まとめると, $I - W^{-\frac{1}{2}}DW^{-\frac{1}{2}}$ の固有値問題を解いて Fielder ベクトルを求める。Fielder ベクトルの正負でデータを 2 つのクラスタに分割する。そして, クラスタ内の類似度平均が小さいクラスタをさらに 2 分割して再帰的に固有値問題を解いて目的のクラスタ数を得る。これが Mcut である。

Mcut のアルゴリズムを以下に示す。

Mcut

1. データ全体で初期クラスタを形成する
2. 各クラスタの類似度平均を計算し, 一番低いクラスタを分割対象クラスタとする
3. 分割対象クラスタの $L = I - W^{-\frac{1}{2}}DW^{-\frac{1}{2}}$ を求める
4. L の固有値問題を解いて Fielder ベクトルを求める
5. Fielder ベクトルの正負でクラスタを 2 分割する。
6. 目標のクラスタ数が得られたなら終了し、そうでなければステップ 2 へ

以下に例を示す。

k-means の例で使った 7 個の 3 次元データ $x_i(i = 1 \cdots 7)$ を Mcut で 3 つのクラスタに分割する。データ行列 X は以下の通りである。

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \begin{pmatrix} 5 & 5 & 5 \\ 9 & 2 & 3 \\ 2 & 2 & 3 \\ 4 & 8 & 9 \\ 8 & 7 & 8 \\ 2 & 9 & 4 \\ 4 & 4 & 4 \end{pmatrix}$$

正規化して積をとると X の類似度行列 W は以下のような対称行列となる。小数点第三位まで表示する。

$$W = \begin{pmatrix} 1.000 & 0.833 & 0.980 & 0.955 & 0.998 & 0.861 & 1.000 \\ 0.833 & 1.000 & 0.775 & 0.642 & 0.852 & 0.492 & 0.833 \\ 0.980 & 0.775 & 1.000 & 0.974 & 0.984 & 0.820 & 0.980 \\ 0.955 & 0.642 & 0.974 & 1.000 & 0.947 & 0.909 & 0.955 \\ 0.998 & 0.852 & 0.984 & 0.947 & 1.000 & 0.830 & 0.998 \\ 0.861 & 0.492 & 0.820 & 0.909 & 0.830 & 1.000 & 0.861 \\ 1.000 & 0.833 & 0.980 & 0.955 & 0.998 & 0.861 & 1.000 \end{pmatrix}$$

W から D は以下の行列となる。

$$D = \begin{pmatrix} 6.629 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 5.430 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 6.515 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 6.385 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 6.611 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 5.776 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 6.629 \end{pmatrix}$$

W と D から正規化ラプラス行列 L は以下の行列となる。

$$L = \begin{pmatrix} 0.849 & -0.138 & -0.149 & -0.146 & -0.150 & -0.139 & -0.150 \\ -0.138 & 0.815 & -0.130 & -0.109 & -0.142 & -0.087 & -0.138 \\ -0.149 & -0.130 & 0.846 & -0.151 & -0.149 & -0.133 & -0.149 \\ -0.146 & -0.109 & -0.151 & 0.843 & -0.145 & -0.149 & -0.146 \\ -0.150 & -0.142 & -0.149 & -0.145 & 0.848 & -0.134 & -0.150 \\ -0.139 & -0.0879 & -0.133 & -0.149 & -0.134 & 0.826 & -0.139 \\ -0.150 & -0.138 & -0.149 & -0.146 & -0.150 & -0.139 & 0.849 \end{pmatrix}$$

L の固有値問題を解いて Fielder ベクトル f を得る。

$$f = \begin{pmatrix} 0.030 & 0.760 & -0.017 & -0.289 & 0.087 & -0.573 & 0.030 \end{pmatrix}$$

f の正負でクラスタを 2 分割するので, X は以下のクラスタ C_1, C_2 に分割される。

$$C_1 = \{x_1, x_2, x_5, x_7\} \;\; C_2 = \{x_3, x_4, x_6\}$$

各クラスタの類似度平均を求める。C_1 の類似度平均は 0.939, C_2 の類似度平均は 0.934 なので, C_2 のほうが類似度平均が低い。よって次に 2 分割するクラスタは C_2 である。

C_2 に属するデータに対応する X の部分行列 X_2 は以下の通りである。

$$X_2 = \begin{pmatrix} x_3 \\ x_4 \\ x_6 \end{pmatrix} = \begin{pmatrix} 2 & 2 & 3 \\ 4 & 8 & 9 \\ 2 & 9 & 4 \end{pmatrix}$$

後は先程と同じようにして X_2 の正規化ラプラス行列 L_2 を求めると

$$L_2 = \begin{pmatrix} 0.642 & -0.343 & -0.297 \\ -0.343 & 0.653 & -0.324 \\ -0.297 & -0.324 & 0.633 \end{pmatrix}$$

L_2 の固有値問題を解いて Fielder ベクトル f_2 を得る。

$$f_2 = \begin{pmatrix} 0.600 & 0.168 & -0.781 \end{pmatrix}$$

f_2 より X_2 は以下のクラスタ C_2, C_3 に分割される。

$$C_2 = \{x_3, x_4\} \ \ C_3 = \{x_6\}$$

目標のクラスタ数が得られたのでここで終了する。最終的なクラスタリング結果は以下の通りである。

$$C_1 = \{x_1, x_2, x_5, x_7\} \ \ C_2 = \{x_3, x_4\} \ \ C_3 = \{x_6\}$$

4.6 スペクトラル k-means

Mcut の手法を用いたスペクトラルクラスタリングでは, 目標のクラスタ数を k とすると $(k-1)$ 回の固有値問題を解かなくてはならない。そのため計算コストが大幅に大きくなる。そこで, 一回の固有値問題でクラスタリング結果を得る手法 [3, 4] を紹介する。この手法を便宜的にスペクトラル K-menas と呼ぶことにする。

スペクトラル k-means は Mcut の手法の固有値問題を解くところまでは同じである。固有値問題を解いて, 小さい方から k 個取り出した固有値に対応する固有ベクトルを取り出す。それらを行列の列として新たな行列を作り, 行を大きさ 1 に正規化する。これは入力元のデータ行列を k 次元に縮約した形となる。そして, この行列を k-means を用いて k 個のクラスタに分割する。このようにして 1 回の固有値問題でクラスタリングを行うことができる。

スペクトラル k-means のアルゴリズムを以下に示す。

スペクトラル k-means

データ行列を $X \in R^{n \times m}$, 目標のクラスタ数を k とする

1. データ行列 X でクラスタを形成する
2. $L = I - W^{-\frac{1}{2}} D W^{-\frac{1}{2}}$ を求める
3. L の固有値問題を解く
4. 小さい方から k 個取り出した固有値に対応する固有ベクトルを $v_1, \cdots, v_k$ とする
5. $v_1, \cdots, v_k$ を列とした行列を $V \in R^{n \times k}$ とする
6. V の行を大きさ 1 に正規化した行列を $U \in R^{n \times k}$ とする
7. U を新たなデータ行列として, kkz+k-means で k 個のクラスタに分割する

以下に例を示す。

7 個の 5 次元データ $x_i(i = 1 \cdots 7)$ をスペクトラル k-means で 3 つのクラスタに分割する。データ行列 X は以下の通りである。

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \begin{pmatrix} 5 & 5 & 5 & 4 & 0 \\ 9 & 2 & 3 & 0 & 2 \\ 2 & 2 & 3 & 3 & 9 \\ 4 & 8 & 9 & 0 & 0 \\ 8 & 7 & 8 & 8 & 1 \\ 2 & 9 & 4 & 2 & 0 \\ 4 & 4 & 4 & 5 & 3 \end{pmatrix}$$

正規化ラプラス行列 L は以下のようになる。

$$L = \begin{pmatrix} 0.829 & -0.140 & -0.098 & -0.155 & -0.169 & -0.154 & -0.158 \\ -0.140 & 0.789 & -0.109 & -0.125 & -0.138 & -0.095 & -0.131 \\ -0.098 & -0.109 & 0.750 & -0.084 & -0.111 & -0.083 & -0.154 \\ -0.155 & -0.125 & -0.084 & 0.811 & -0.145 & -0.170 & -0.131 \\ -0.169 & -0.138 & -0.111 & -0.145 & 0.828 & -0.145 & -0.164 \\ -0.154 & -0.095 & -0.083 & -0.170 & -0.145 & 0.805 & -0.138 \\ -0.158 & -0.131 & -0.154 & -0.131 & -0.164 & -0.138 & 0.827 \end{pmatrix}$$

L の固有値問題を解く。ここまでは Mcut と同じである。固有値が小さい順に対応する固有ベクトルをクラスタ分割数だけ取り出す。よって固有ベクトルを 3 つ取り出し, v_1, v_2, v_3 とし, 以下のように固有ベクトルを列とする行列を作る。

$$V = \begin{pmatrix} v_1 & v_2 & v_3 \end{pmatrix} = \begin{pmatrix} -0.399 & 0.180 & 0.055 \\ -0.359 & -0.090 & 0.837 \\ -0.330 & -0.823 & -0.294 \\ -0.380 & 0.337 & -0.118 \\ -0.398 & 0.081 & 0.042 \\ -0.374 & 0.365 & -0.427 \\ -0.397 & -0.163 & -0.096 \end{pmatrix}$$

V の行を大きさ 1 に正規化した行列 U は以下の行列となる。

$$U = \begin{pmatrix} -0.903 & 0.408 & 0.126 \\ -0.392 & -0.099 & 0.914 \\ -0.353 & -0.880 & -0.315 \\ -0.728 & 0.646 & -0.226 \\ -0.974 & 0.199 & 0.105 \\ -0.554 & 0.540 & -0.632 \\ -0.902 & -0.370 & -0.218 \end{pmatrix}$$

このように U は X を 3 次元に縮約した行列となっている。最後に kkz+k-means で U を 3 つのクラスタに分割すると最終的なクラスタリング結果は以下のようになる。

$$C_1 = \{x_3, x_7\} \;\; C_2 = \{x_1, x_4, x_5, x_6\} \;\; C_3 = \{x_2\}$$

第5章　アンサンブルクラスタリング

5.1　アンサンブルクラスタリング

複数のクラスタリング結果を用いて, 新たなデータ行列を作り, そのデータ行列でさらにクラスタリングを行う手法がアンサンブルクラスタリング [2] である。K-menas などのように初期値やパラメータでクラスタリング結果にバラツキがある場合に, それらの結果を平均的な結果を与えるのに有効な手法である。通常は, アンサンブルされる各クラスタリング結果よりもよいクラスタリングを得られる。

アンサンブルクラスタリングは, 複数のクラスタリング結果からどのようにして新たな行列を作るかが重要である。ここではハイパーグラフという手法を用いて行列を作る。この行列を kkz+k-means を用いて k 個のクラスタに分割する。

5.2　ハイパーグラフ

ハイパーグラフ [2] について説明する。データ数を n, クラスタ分割数を k, クラスタリング結果の数を y とする。クラスタリング結果からハイパーグラフを構成する部分行列 $V \in R^{n \times k}$ を作る。V の行は各データに対応しており, 列は各クラスタに対応している。V_{ij} は、i 番目のデータが j 番目のクラスタに属していれば 1 となり、そうでなければ 0 である。このようにしてそれぞれのクラスタリング結果から部分行列を作っていき, できあがった y 個の部分行列を列単位で結合することでハイパーグラフは完成する。よってハイパーグラフは n 行 yk 列の行列となる。

以下にハイパーグラフ作成例を示す。

データ数を 5, クラスタ分割数を 2, そして 3 つのクラスタリング結果を得たとする。
クラスタリング結果は

$$y_1 = (1, 2, 1, 2, 1) \quad y_2 = (1, 1, 2, 2, 2) \quad y_3 = (2, 2, 2, 2, 1)$$

である。この結果からハイパーグラフを構成する部分行列は

$$V_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \quad V_2 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} \quad V_3 = \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$

となる。

V_1, V_2, V_3 を結合すると, ハイパーグラフ H は

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

となる。

第6章　実験

6.1　実験内容

クラスタ分割数を k とすると, スペクトラル k-means は k 個の正規化ラプラス行列の固有ベクトルで新たな行列を作る。これは縮約次元数を k として, 入力行列を k 次元に縮約しているということである。ここで縮約次元数を変えることにより, クラスタリング結果に多様性が出る。そこで, 縮約次元数を色々と変えて得た複数のスペクトラル k-means の結果を用いてアンサンブルクラスタリングを行えばどのような結果となるか実験する。

実験では R というフリーの統計解析ソフトを使って, Mcut, k-means, スペクトラル k-means, 本手法の 4 つの手法の比較を行う。本手法においてアンサンブルさせるスペクトラル k-means の結果は縮約次元数を $(k \times 1.0)$, $(k \times 1.2)$, $(k \times 1.4)$, $(k \times 1.6)$, $(k \times 1.8)$, $(k \times 2.0)$ と変えた 6 つのパターンでの結果を用いる。クラスタリング結果はエントロピーと純度で評価する。

6.2　データセット

実験に利用するデータセットは CLUTO というクラスタリングツールのテスト用データを使った。このデータは以下の URL からダウンロードできる。

http://glaros.dtc.umn.edu/gkhome/views/cluto

データセットの詳細を表 6.1 に示す。

表 6.1: データセット

	データ数	次元数	クラスタ数
fbis	2463	2000	17
re0	1504	2886	13
re1	1657	3758	25
tr11	414	6429	9
tr12	313	5804	8
tr23	204	5832	6
tr31	927	10128	7
tr41	878	7454	10
tr45	690	8261	10
wap	1560	8460	20

実際に使う場合は, スパース行列の形式となっているデータセットのファイルを R で読み込んで通常の行列に変換して, さらに行の大きさを 1 に正規化した行列をクラスタリングに使用する。

6.3 実験結果

縮約次元数が $(k \times 1.0)$, $(k \times 1.2)$, $(k \times 1.4)$, $(k \times 1.6)$, $(k \times 1.8)$, $(k \times 2.0)$ の場合のスペクトラル k-means のクラスタリング結果のエントロピーと純度をそれぞれ表 6.2, 表 6.3 に示す。これらのクラスタリング結果は本手法に用いる。

各データセットで最も結果が良い値を太字で示す。

表 6.2: スペクトラル k-means のエントロピー

	$(k \times 1.0)$	$(k \times 1.2)$	$(k \times 1.4)$	$(k \times 1.6)$	$(k \times 1.8)$	$(k \times 2.0)$
fbis	0.377	0.373	0.379	0.365	**0.342**	0.349
re0	0.411	0.402	0.412	0.403	**0.398**	0.407
re1	0.395	**0.375**	0.387	0.381	0.396	0.387
tr11	0.304	0.295	0.288	0.334	0.295	**0.277**
tr12	0.379	0.350	0.336	0.359	**0.323**	0.340
tr23	0.561	0.512	**0.508**	0.565	0.664	0.645
tr31	0.467	0.412	0.418	0.267	0.268	**0.216**
tr41	0.283	0.309	0.255	**0.207**	0.353	0.397
tr45	0.351	0.338	0.356	**0.187**	0.307	0.296
wap	0.362	**0.359**	0.364	0.388	0.382	0.393
平均	0.389	0.372	0.370	**0.346**	0.373	0.371

表 6.3: スペクトラル k-means の純度

	$(k \times 1.0)$	$(k \times 1.2)$	$(k \times 1.4)$	$(k \times 1.6)$	$(k \times 1.8)$	$(k \times 2.0)$
fbis	0.658	0.644	0.622	0.652	**0.679**	0.652
re0	0.632	**0.644**	0.606	0.638	0.643	0.632
re1	0.606	**0.629**	0.609	0.608	0.619	0.612
tr11	0.749	0.756	0.761	0.691	**0.768**	0.761
tr12	0.712	0.732	0.722	0.728	**0.751**	0.732
tr23	0.627	0.691	**0.701**	0.613	0.500	0.525
tr31	0.619	0.686	0.695	**0.835**	0.799	0.828
tr41	0.786	0.713	0.798	**0.844**	0.702	0.656
tr45	0.729	0.743	0.710	**0.884**	0.745	0.768
wap	0.666	**0.676**	0.652	0.617	0.641	0.622
平均	0.678	0.691	0.688	**0.711**	0.685	0.679

表 6.2 の各縮約次元数によるエントロピーの平均を図 6.2 に示す。

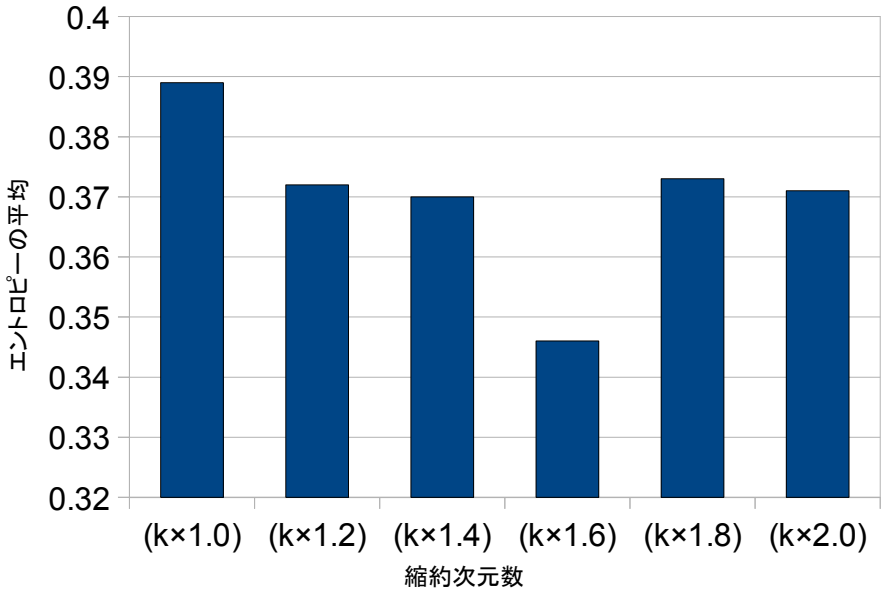

図 6.2: 各縮約次元数のエントロピーの平均

表 6.3 の各縮約次元数による純度の平均を図 6.3 に示す。

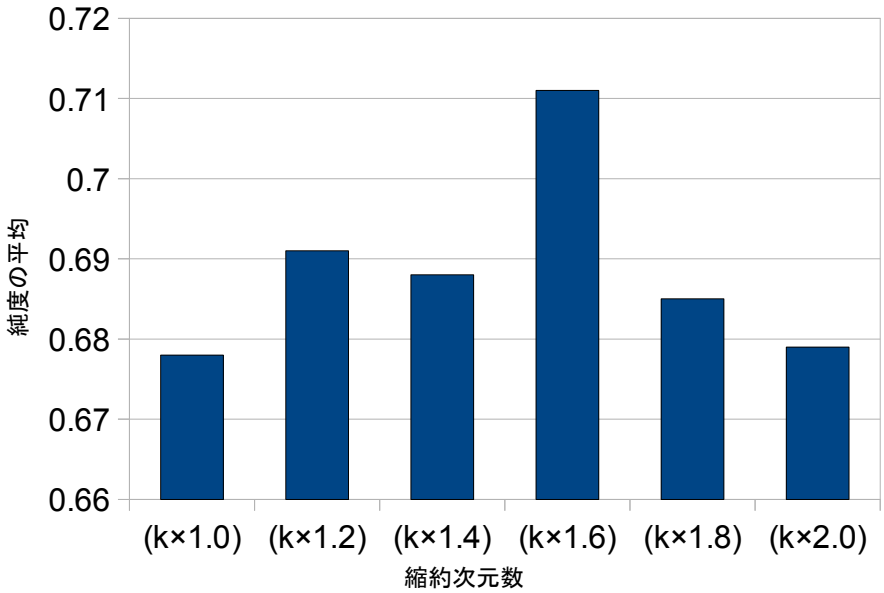

図 6.3: 各縮約次元数の純度の平均

各手法のクラスタリング結果のエントロピーと純度をそれぞれ表 6.4, 表 6.5 に示す。スペクトラル k-means を Sk-means と表記する。

各データセットで最も結果が良い値を太字で示す。

表 6.4: 各手法のエントロピー

	Mcut	k-means	Sk-means	本手法
fbis	0.431	**0.335**	0.377	0.360
re0	0.508	0.405	0.411	**0.400**
re1	0.466	0.386	0.395	**0.376**
tr11	0.476	**0.272**	0.304	0.274
tr12	0.696	0.438	0.379	**0.352**
tr23	0.661	**0.505**	0.561	0.541
tr31	0.538	0.400	0.467	**0.327**
tr41	0.575	0.361	0.283	**0.241**
tr45	0.623	0.427	0.351	**0.239**
wap	0.498	0.409	0.362	**0.350**
平均	0.547	0.394	0.389	**0.346**

表 6.5: 各手法の純度

	Mcut	k-means	Sk-means	本手法
fbis	0.590	**0.694**	0.658	0.658
re0	0.561	0.612	0.632	**0.648**
re1	0.556	**0.625**	0.606	0.622
tr11	0.626	0.742	0.749	**0.768**
tr12	0.419	0.668	0.712	**0.735**
tr23	0.520	**0.667**	0.627	0.657
tr31	0.633	0.691	0.619	**0.735**
tr41	0.500	0.705	0.786	**0.808**
tr45	0.442	0.625	0.729	**0.851**
wap	0.499	0.587	0.666	**0.682**
平均	0.535	0.662	0.678	**0.716**

表 6.4 の各手法によるエントロピーの平均を図 6.4 に示す。

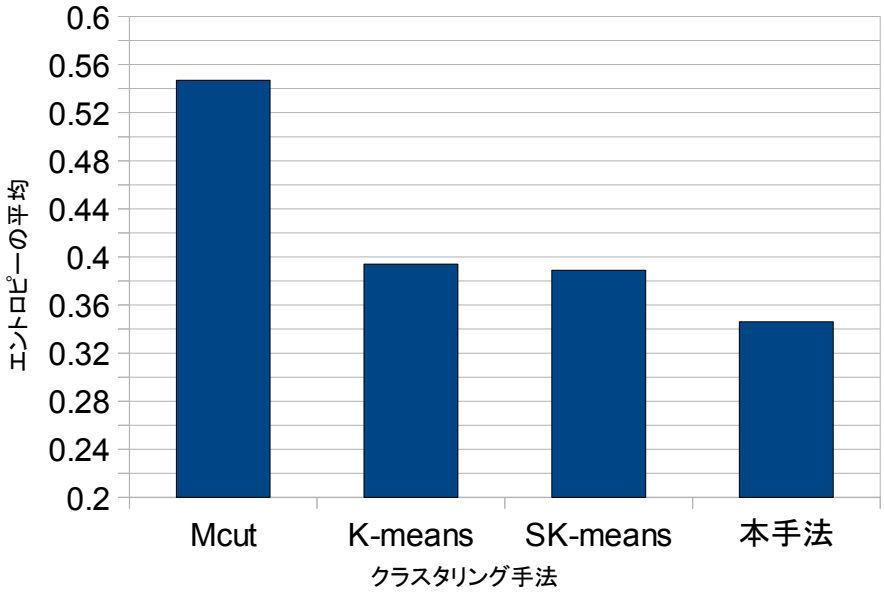

図 6.4: 各手法のエントロピーの平均

表 6.5 の各手法による純度の平均を図 6.5 に示す。

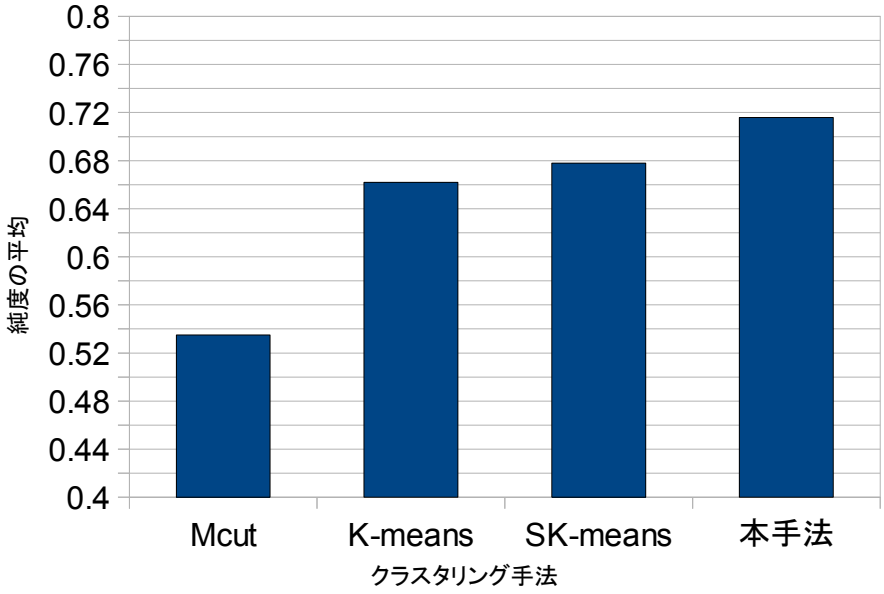

図 6.5: 各手法の純度の平均

エントロピー・純度ともにほとんどのデータセットで本手法が一番良い結果が出た。スペクトラル k-means では, 通常のスペクトラル k-means と同じである縮約次元数をクラスタ分割数の 1.0 倍に設定した場合よりも, 他の倍率のほうが結果が良い。本手法はそれらをアンサンブルクラスタリングにかけたので, 通常のスペクトラル k-means よりも精度が良くなったのである。また, 平均を比較してみるとスペクトラル k-means のほうが k-means より結果が良くなっていることがわかる。

6.4 考察

データセット tr31 では, スペクトラル k-means より k-means のほうが明らかに結果良く, エントロピー・純度の差がはっきりと出ている。これはスペクトラル k-means でうまく次元を縮約できなかったことが原因である。スペクトラル k-means はスペクトラルクラスタリングに分類されているが, 結局のところ, 主成分分析や plsi のような次元縮約法でデータを低次元に縮約し, k-means にかけてるだけであり, クラスタリング自体は k-means が行っているといえる。tr31 の次元数は他のデータセットと比べて明らかに多いし, クラスタ分割数も 7 と少ない。このデータをスペクトラル k-means では, 7 次元に縮約しているので, 次元縮約によるデータ情報の欠損が大きかったのではないかと考えられる。tr31 の縮約次元数を増やしていけば, エントロピー・純度の数値が良くなっていることからもそれがわかる。そして, このデータ情報の欠損が後処理である k-means のクラスタリング結果に影響が出たといえる。ただし, これは tr31 だけに着目していえることである。なぜなら, tr31 では縮約次元数を増やしていけば, エントロピー・純度が良くなっているが, 他のデータセットではそのような傾向になっていないからである。よって, 他のデータでは次元縮約による情報欠損がそんなに大きくないものと考えられる。

逆に, データセット tr41, tr45 では, スペクトラル k-means のほうが k-means より明らかに結果が良くなっている。k-means は線形分離不可能なデータに対してクラスタリングがうまく行えない。しかし, スペクトラル k-means では次元縮約を行い低次元空間でデータを表現することにより線形分離可能な新たなデータを作る。このデータを k-means にかけるので, 素のデータを直接 k-means にかけてやるより良いクラスタリング結果となる。このことから, スペクトラル k-means のほうが k-means より良い結果である原因は, tr41 と tr45 が線形分離不可能なデータであったと考えることができる。

本手法の精度を上げるための改善点を考える。実験では縮約次元数が 6 パターンでの結果をアンサンブルクラスタリングに使用したが, その 6 パターンでのスペクトラル k-means の結果が最も良いものとは限らない。よって, アンサンブルさせるスペクトラル k-means のクラスタリング結果を増やせば結果が良くなるのではないかと考えられる。また, スペクトル k-means の類似度行列の作り方を改善すれば, それが結果的に本手法の改善につながるのではないかと考える。類似度行列は内積で簡単に求めたが, このやり方は計算コストは非常に優れているが, 精度は悪い。よって時間はかかるがユークリッド距離で類似度行列を作れば, 内積で作った類似度行列より精度の高い類似度行列ができ, その結果, スペクトラル K-menas の精度も上がるのではないかと考えられる。最終的にはその結果をアンサンブルさせるので, 本手法の精度の改善につながる。

第7章　結論

本論文では, スペクトラル K-means の縮約次元数によるクラスタリング結果の多様性に着目して, 平均的に精度の高いクラスタリングを行うをこと目的に, 縮約次元数を色々と変えてスペクトラル K-means を実行した複数のクラスタリング結果を用いてアンサンブルクラスタリングを行う手法を提案した。実験では, K-means, スペクトラルクラスタリング及びスペクトラル K-means と本手法の比較実験を行い, 本手法の有効性を示した。今後の課題としては, アンサンブルさせるスペクトラル K-means の次元数の設定をより明確に導き出し, アンサンブルさせるクラスタリング結果の数を増やして, より精度の高いクラスタリングを実現していく必要がある。

謝 辞

本研究の遂行及び論文の作成に多大なご助言及び指導を賜った新納浩幸准教授に深い感謝の意を表します。

参 考 文 献

[1] 新納浩幸,『R で学ぶクラスタ解析』, オーム社, 2007.

[2] 新納浩幸/佐々木稔,「NMF による重み付きハイパーグラフを用いたアンサンブル文書クラスタリング」, 2006.

[3] Ulrike von Luxburg, A Tutorial on Spectral Clustering, 2006.

[4] Andrew Y. Ng, Michael I. Jordan and Yair Weiss, On Spectral Clustering:Analysis and an algorithm, 2001.

[5] Ji Hey, Man Lanz, Chew-Lim Tanz, Sam-Yuan Sungz and Hwee-Boon Low, Initialization of Cluster Refinement Algorithms:A Review and Comparative Study, 2005

付 録A　サンプルプログラム

- Mcut

```
mcut <- function(filename,k) {
    X <- readMM(filename)
    X <- seikika(X)
    W <- X %*% t(X)
    ansv <- numeric(nrow(W)) + 1 #クラスタリング結果のベクトル
    nowc <- 1 #分割するクラス番号
    nowk <- 1 #現在のクラス数
    simv <- simave(W) #類似度平均ベクトル
    mcutcore(W, k, ansv, nowc, nowk, simv)
}
mcutcore <- function(W, k, ansv, nowc, nowk, simv) {
    #クラス nowc に属するデータの類似度行列作成
    Wsub <- matsub(W, ansv, nowc)
    #分割
    dg <- apply(Wsub, 1, sum)
    D <- diag(dg)
    I <- diag( ncol(D) )
    D <- diag( 1 / sqrt(dg) )
    X <- I - ( D %*% Wsub %*% D )
    ans <- eigen(X)
    fv <- ans$vectors[,order(ans$values)[2]]
    q <- D %*% fv
    q[q >= 0] <- nowc
    nowk <- nowk + 1
    q[q < 0] <- nowk
    ansv[ansv == nowc] <- q
    #類似度平均ベクトル更新
    simv[nowc] <- simave( matsub(W, ansv, nowc) )
    simv <- c( simv, simave(matsub(W, ansv, nowk)) )
    #類似度平均の一番低いクラスを分割
    nowc <- order(simv)[1]
    if(k > nowk) { ansv <- mcutcore(W, k, ansv, nowc, nowk ,simv) }
    return(ansv)
}
#類似度平均計算
simave <- function(W) {
    n <- nrow(W)
    ave <- sum( rowSums(W) ) / (n * n)
    return(ave)
}
#部分行列作成
matsub <- function(W, ansv, nowc) {
    index <- which(ansv == nowc)
    return(W[index, index])
}
```

- スペクトラル k-means

```
skmeans <- function(filename,k) {
    X <- readMM(filename)
    X <- seikika(X)
    W <- X %*% t(X)
    dg <- apply(W, 1, sum)
    D <- diag(dg)
    I <- diag( ncol(D) )
    L <- I - diag(1/sqrt(dg)) %*% W %*% diag(1/sqrt(dg))
    el <- eigen(L)
    V <- el$vectors[,order(el$values)]
    V <- V[,c(1:(k))]
    U <- V / sqrt( rowSums(V*V) )
    K <- kkz(U,k)
    km <- kmeans(U,K)
    km$cluster
}
```

- kkz

```
kkz <- function(X, k) {
    norm <- apply(X^2,1,sum)
    cv <- order(norm, decreasing=TRUE)[1]
    C <- t(X[cv,])
    while( length(cv) < k ) {
        dmax <- apply(X, 1, function(x) {
                                dmin <- rowSums(t(t(C)-x)^2)
                                return( min(dmin) )
                            })
        cv <- c( cv, order(dmax, decreasing=TRUE)[1] )
        C <- X[cv,]
    }
    return(C)
}
```

- アンサンブルスペクトラルクラスタリング

```
encl <- function(filename, k) {
    X <- readMM(filename)
    X <- seikika(X)
    W <- X %*% t(X)
    dg <- apply(W, 1, sum)
    D <- diag(dg)
    I <- diag( ncol(D) )
    L <- I - diag(1/sqrt(dg)) %*% W %*% diag(1/sqrt(dg))
    el <- eigen(L)
    V <- el$vectors[,order(el$values)]
    ans <- spcl(V, k, 1.0)
    X1 <- matrix(0, nrow=length(ans), ncol=k)
    for (i in 1:length(ans)) { X1[i, ans[i]] <- 1 }
    ans <- spcl(V, k, 1.2)
    X2 <- matrix(0, nrow=length(ans), ncol=k)
    for (i in 1:length(ans)) { X2[i, ans[i]] <- 1 }
    ans <- spcl(V, k, 1.4)
    X3 <- matrix(0, nrow=length(ans), ncol=k)
    for (i in 1:length(ans)) { X3[i, ans[i]] <- 1 }
    ans <- spcl(V, k, 1.6)
    X4 <- matrix(0, nrow=length(ans), ncol=k)
    for (i in 1:length(ans)) { X4[i, ans[i]] <- 1 }
    ans <- spcl(V, k, 1.8)
    X5 <- matrix(0, nrow=length(ans), ncol=k)
    for (i in 1:length(ans)) { X5[i, ans[i]] <- 1 }
    ans <- spcl(V, k, 2.0)
    X6 <- matrix(0, nrow=length(ans), ncol=k)
    for (i in 1:length(ans)) { X6[i, ans[i]] <- 1 }
    NEWX <- cbind(X1, X2, X3, X4, X5, X6)
    K <- kkz(NEWX,k)
    km <- kmeans(NEWX,K)
    km$cluster
}
spcl <- function(V ,k ,m) {
    V <- V[,c(1:(k*m))]
    U <- V / sqrt( rowSums(V*V) )
    K <- kkz(U,k)
    km <- kmeans(U,K)
    km$cluster
}
```

- エントロピーと純度

```
myentropy <- function (ct) {
    -sum((rowSums(ct) / sum(ct))
          * apply(ct,1,function(pv) {
                         p1 <- pv/sum(pv)
                         p2 <- p1[p1 != 0]
                         sum(p2 * log(p2))
                       })) /log(ncol(ct))
}
mypurity <- function (ct) {
    sum(apply(ct,1,max)) / sum(ct)
}
myeval <- function(myans, ansfile) {
     goldans <- scan(ansfile,what="character")
     ct <- table(myans,goldans)
     cat("Entropy: ",myentropy(ct),"\n")
     cat("Purity : ",mypurity(ct),"\n")
}
```

- 正規化

```
seikika <- function(X) {
    X2 <- X^2
    norm <- sqrt( rowSums(X2) )
    X <- data.matrix( X / norm )
    return(X)
}
```