

データサイエンスの諸相 (クラスター分析)

前田利之（まえだ としゆき）

maechan@hannan-u.ac.jp

阪南大学 経営情報学部

(2024.11.13)

多次元尺度法（復習）

☆ 【練習の復習】

- 1 前回のデータ I: ¥ rdata ¥ mds01.csv を利用する
<http://puffin.hannan-u.ac.jp/lect/datm/mds01.csv> を利用してもよい

- 2 R を立ち上げ、以下を実行する

```
kan <- read.csv('I:/rdata/mds01.csv',  
  ,header=T,row.names=1) # 本当は1行  
kan1 <- as.dist(kan) # 下の準備  
kan2 <- cmdscale(kan1) # MDS 処理  
plot(kan2,type="n")  
text(kan2,rownames(kan2))
```

クラスター分析

クラスター分析 とは:

- 同じ特徴を持つものをまとめて
 するための手法
- 大きく分けて2種類
 - 似た特徴のものをまとめることを繰り返し、
 構造をつくる
 - 事前に決めた のグループにわける
()

クラスター分析

クラスター分析 (クラスタリング) とは:

- 同じ特徴を持つものをまとめて
[] するための手法
- 大きく分けて2種類
 - 似た特徴のものをまとめることを繰り返し、
[] 構造をつくる
 - 事前に決めた [] のグループにわけ
([])

クラスター分析

クラスター分析 (クラスタリング) とは:

- 同じ特徴を持つものをまとめて
グループを作成するための手法
- 大きく分けて2種類
 - 似た特徴のものをまとめることを繰り返し、
[] 構造をつくる
 - 事前に決めた [] のグループにわけ
([])

クラスター分析

クラスター分析 (クラスタリング) とは:

- 同じ特徴を持つものをまとめて
グループを作成するための手法
- 大きく分けて2種類
 - 似た特徴のものをまとめることを繰り返し、
階層構造をつくる
 - 事前に決めた のグループにわけ
()

クラスター分析

クラスター分析 **(クラスタリング)** とは:

- 同じ特徴を持つものをまとめて **グループを作成** するための手法
- 大きく分けて2種類
 - 似た特徴のものをまとめることを繰り返し、**階層** 構造をつくる
 - 事前に決めた **個数** のグループにわけると
()

クラスター分析

クラスター分析 (クラスタリング) とは:

- 同じ特徴を持つものをまとめて
グループを作成するための手法
- 大きく分けて2種類
 - 似た特徴のものをまとめることを繰り返し、
階層構造をつくる
 - 事前に決めた個数のグループにわけ
(非階層的)

階層的クラスター分析

☆ 距離の近いものをまとめていく。前提として、任意の2点間の距離は計算できるとする

1 距離が最短な2点を選び、それを1つのグループ(点)とする

2 まとめた点と他の点の距離を計算しなおす
() ここで、距離を再計算することで、点が とみなせることに注意

3 上の2ステップを繰り返し、最終的に1点になれば終了

階層的クラスター分析

☆ 距離の近いものをまとめていく。前提として、任意の2点間の距離は計算できるとする

1 距離が最短な2点を選び、それを1つのグループ(点)とする

2 まとめた点と他の点の距離を計算しなおす
(**方法は複数ある**) ここで、距離を再計算することで、点が とみなせることに注意

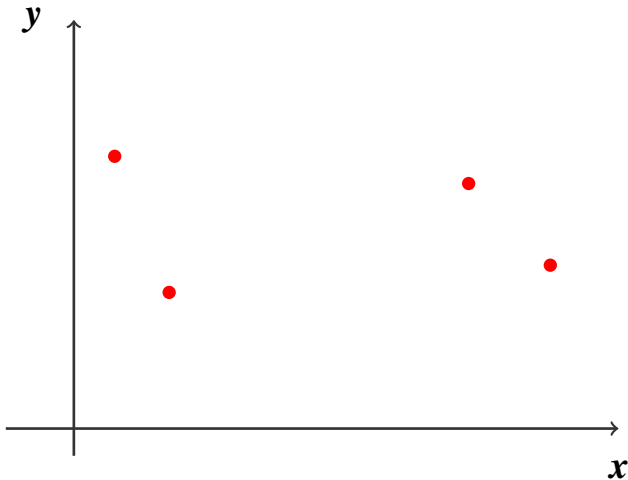
3 上の2ステップを繰り返し、最終的に1点になれば終了

階層的クラスター分析

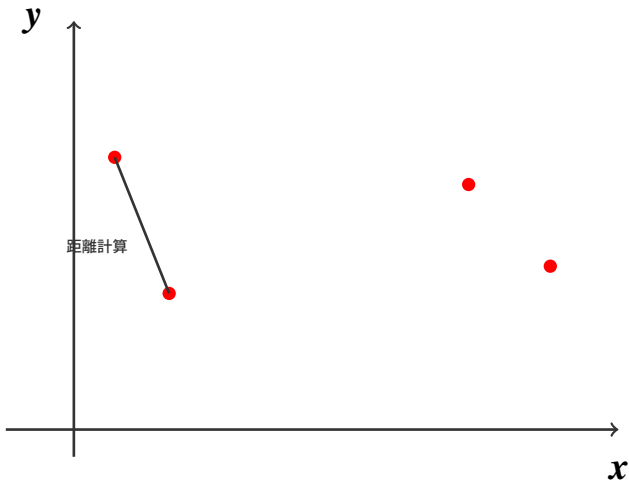
☆ 距離の近いものをまとめていく。前提として、任意の2点間の距離は計算できるとする

- 1 距離が最短な2点を選び、それを1つのグループ(点)とする
- 2 まとめた点と他の点の距離を計算しなおす
(**方法は複数ある**) ここで、距離を再計算することで、点が **1つ減った** とみなせることに注意
- 3 上の2ステップを繰り返し、最終的に1点になれば終了

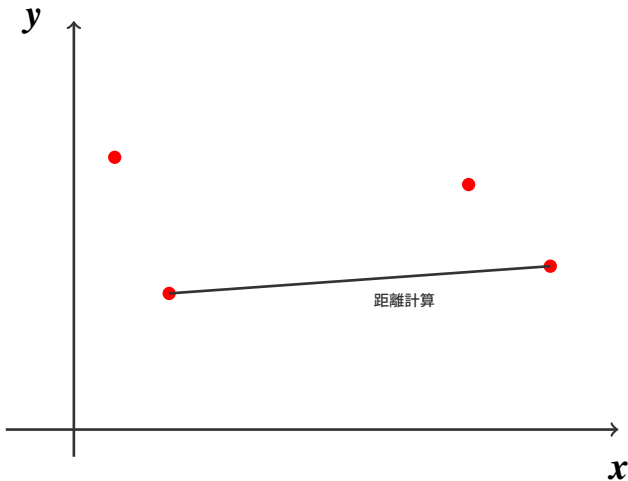
クラスター分析のイメージ



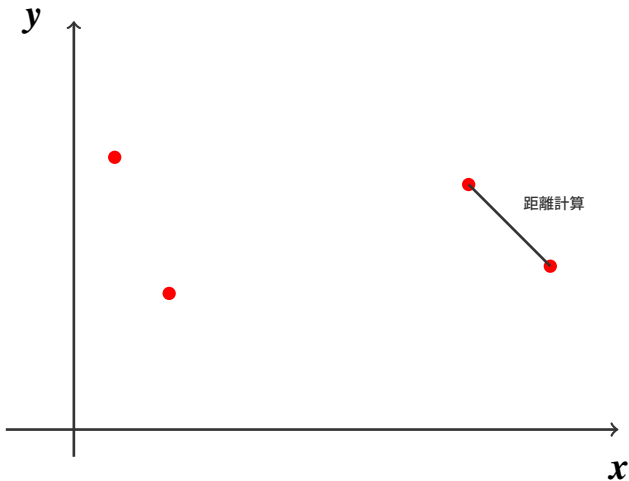
クラスター分析のイメージ



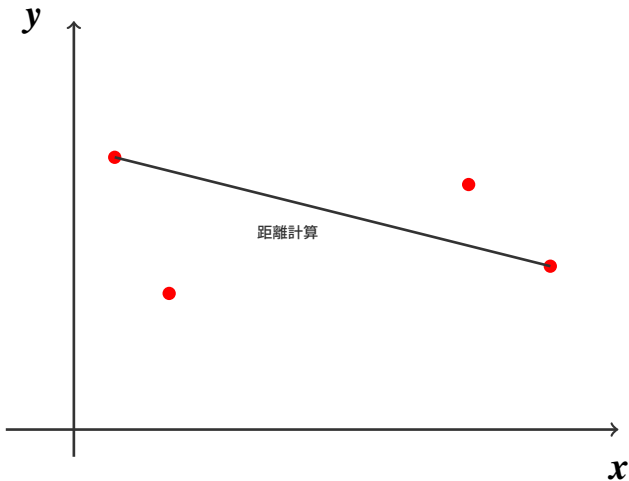
クラスター分析のイメージ



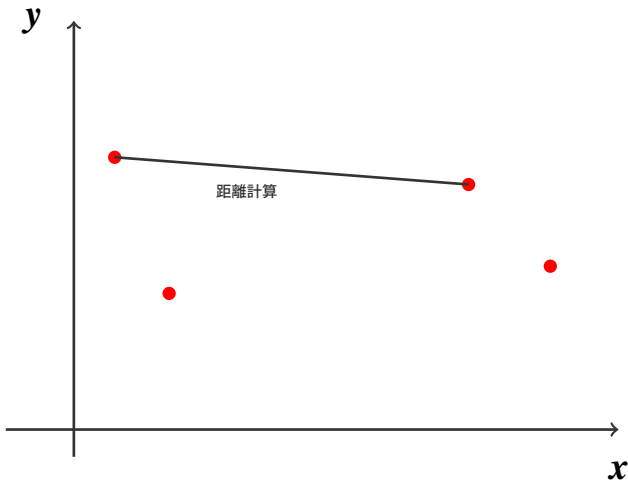
クラスター分析のイメージ



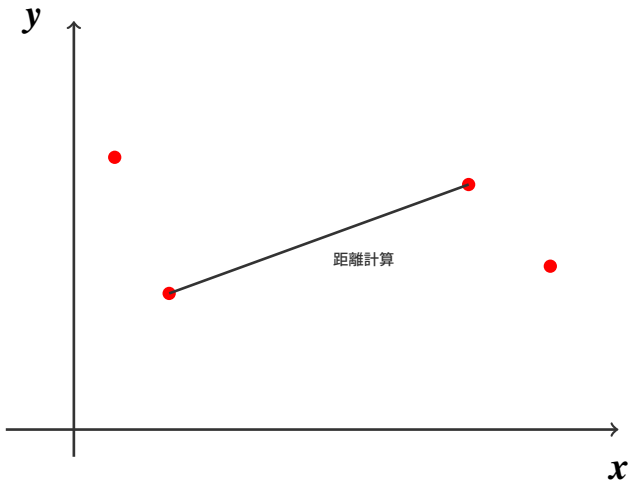
クラスター分析のイメージ



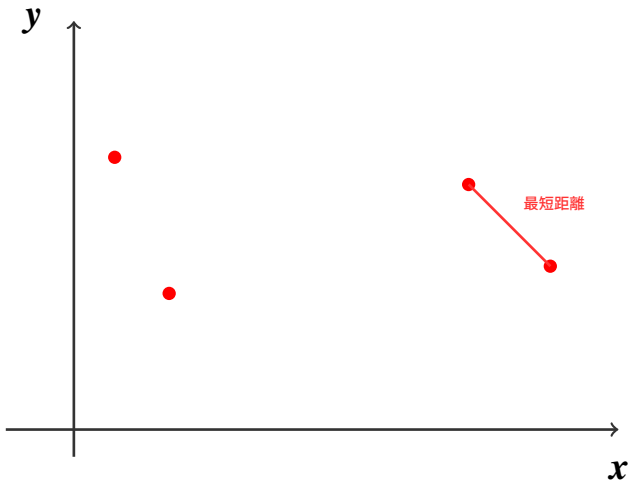
クラスター分析のイメージ



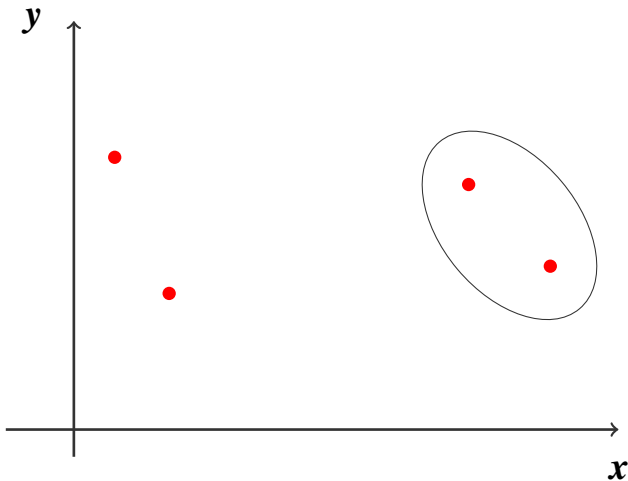
クラスター分析のイメージ



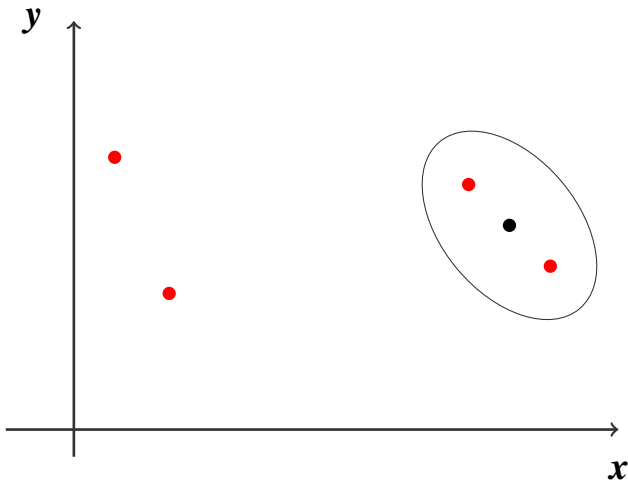
クラスター分析のイメージ



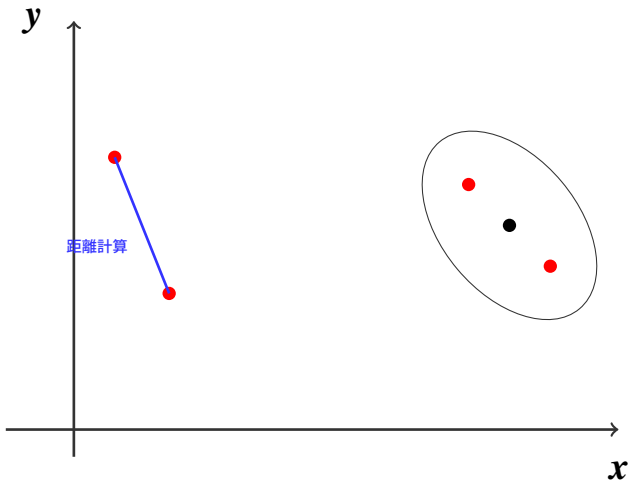
クラスター分析のイメージ



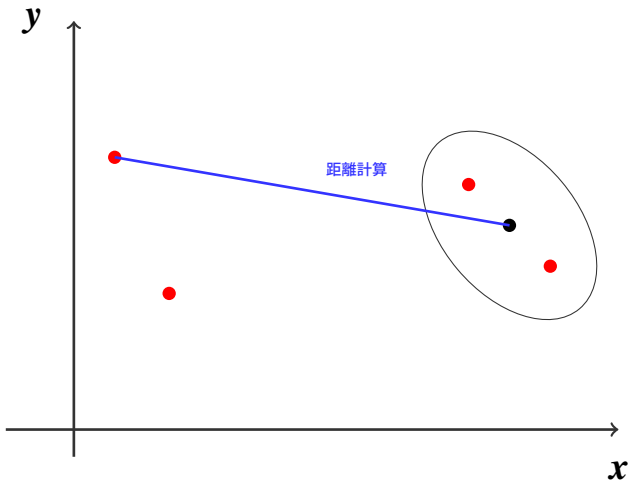
クラスター分析のイメージ



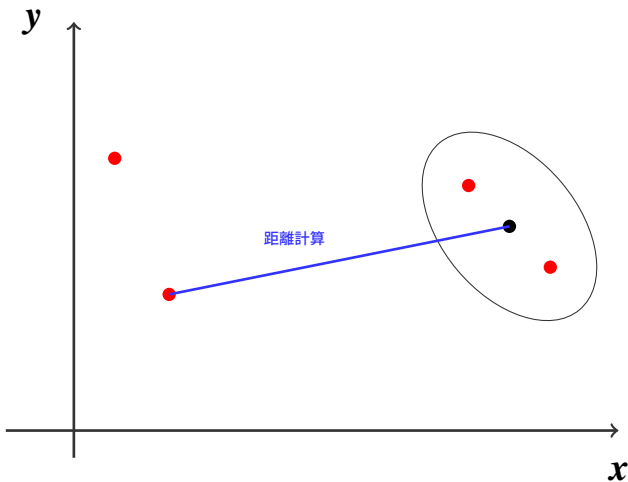
クラスター分析のイメージ



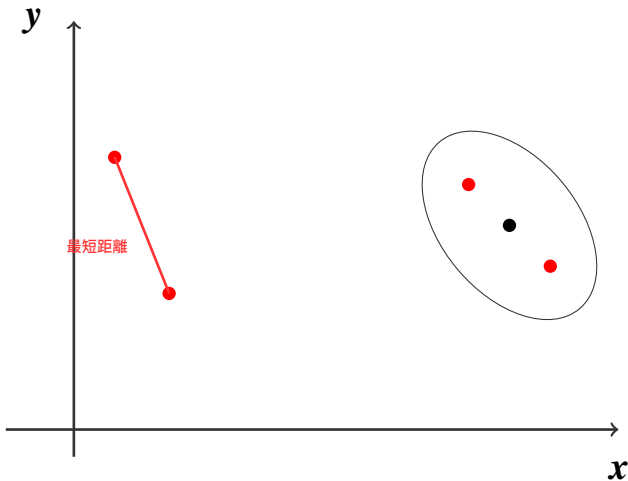
クラスター分析のイメージ



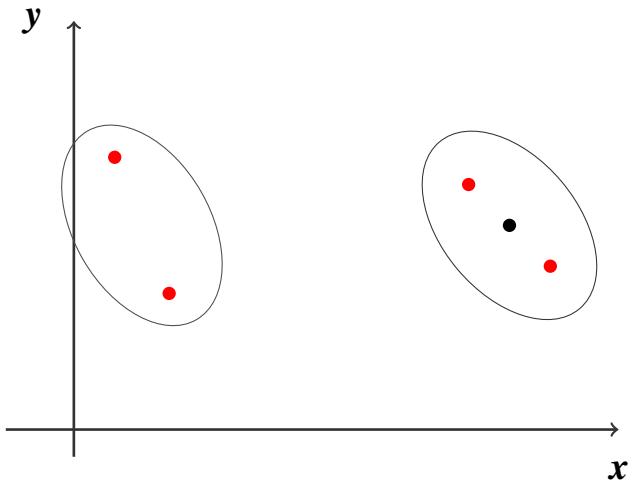
クラスター分析のイメージ



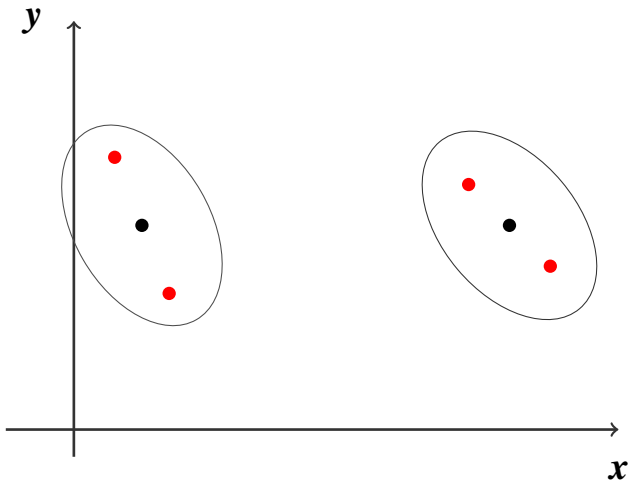
クラスター分析のイメージ



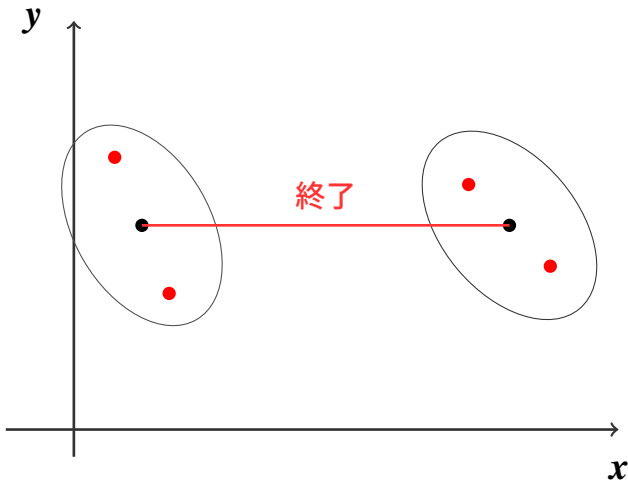
クラスター分析のイメージ



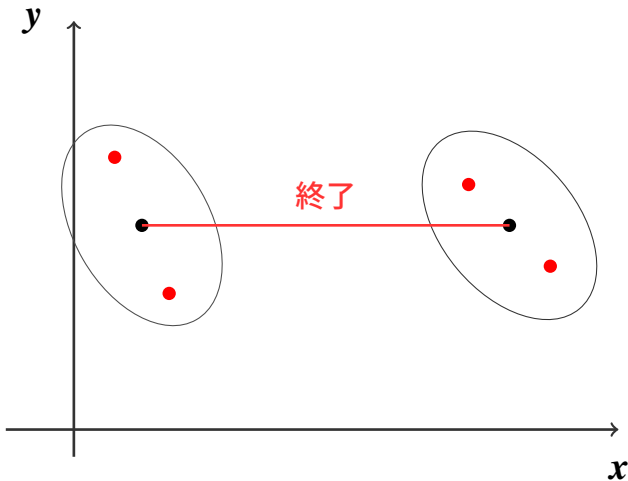
クラスター分析のイメージ



クラスター分析のイメージ



クラスター分析のイメージ



- 2点になればそれをまとめるだけ=終了

グループ間距離の再計算法

- 1 : 一番短い距離を利用する
- 2 : 一番長い距離を利用する
- 3 : 距離の平均を求める
- 4 : 重心との距離を計算する
- 5 : クラスターが巨大にならないようにグループの中の距離がどれだけ増えるかを計算し、それが一番最小となる点を選ぶ

グループ間距離の再計算法

- 1 **最短距離法** : 一番短い距離を利用する
- 2 : 一番長い距離を利用する
- 3 : 距離の平均を求める
- 4 : 重心との距離を計算する
- 5 : クラスタが巨大にならないようにグループの中の距離がどれだけ増えるかを計算し、それが一番最小となる点を選ぶ

グループ間距離の再計算法

- 1 **最短距離法** : 一番短い距離を利用する
- 2 **最長距離法** : 一番長い距離を利用する
- 3 : 距離の平均を求める
- 4 : 重心との距離を計算する
- 5 : クラスタが巨大にならないようにグループの中の距離がどれだけ増えるかを計算し、それが一番最小となる点を選ぶ

グループ間距離の再計算法

- 1 **最短距離法** : 一番短い距離を利用する
- 2 **最長距離法** : 一番長い距離を利用する
- 3 **群平均法** : 距離の平均を求める
- 4 : 重心との距離を計算する
- 5 : クラスタが巨大にならないようにグループの中の距離がどれだけ増えるかを計算し、それが一番最小となる点を選ぶ

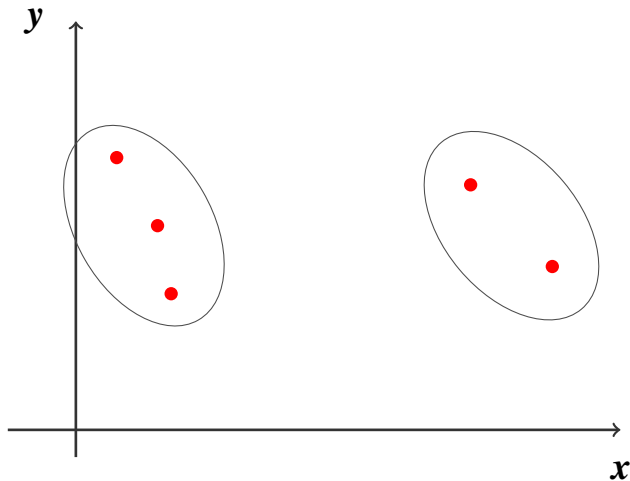
グループ間距離の再計算法

- 1 **最短距離法** : 一番短い距離を利用する
- 2 **最長距離法** : 一番長い距離を利用する
- 3 **群平均法** : 距離の平均を求める
- 4 **重心法** : 重心との距離を計算する
- 5 : クラスタが巨大にならないようにグループの中の距離がどれだけ増えるかを計算し、それが一番最小となる点を選ぶ

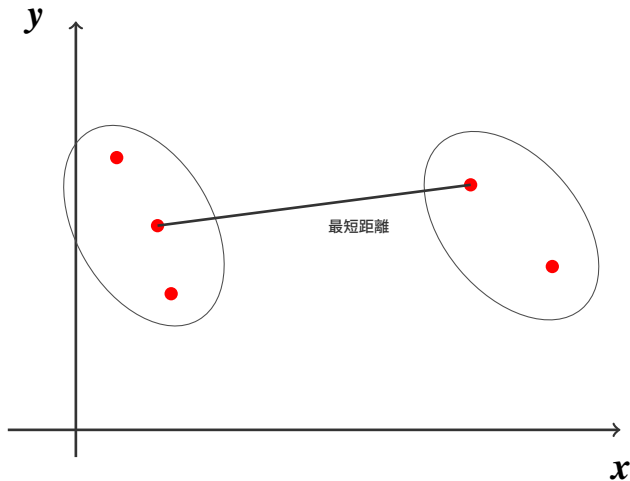
グループ間距離の再計算法

- 1 **最短距離法** : 一番短い距離を利用する
- 2 **最長距離法** : 一番長い距離を利用する
- 3 **群平均法** : 距離の平均を求める
- 4 **重心法** : 重心との距離を計算する
- 5 **ワード法** : クラスタが巨大にならないようにグループの中の距離がどれだけ増えるかを計算し、それが一番最小となる点を選ぶ

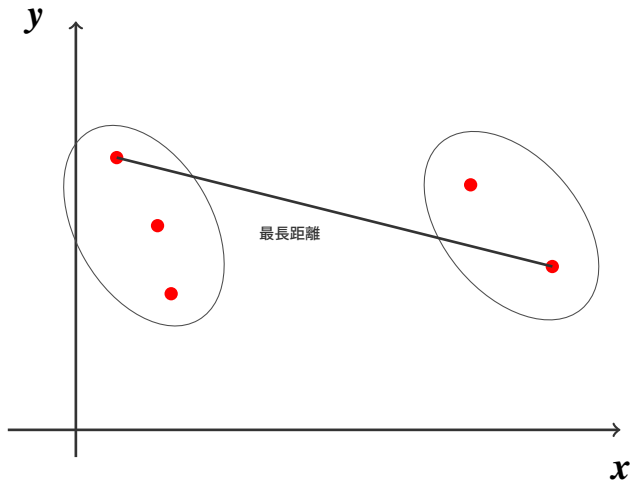
グループ間距離の再計算のイメージ



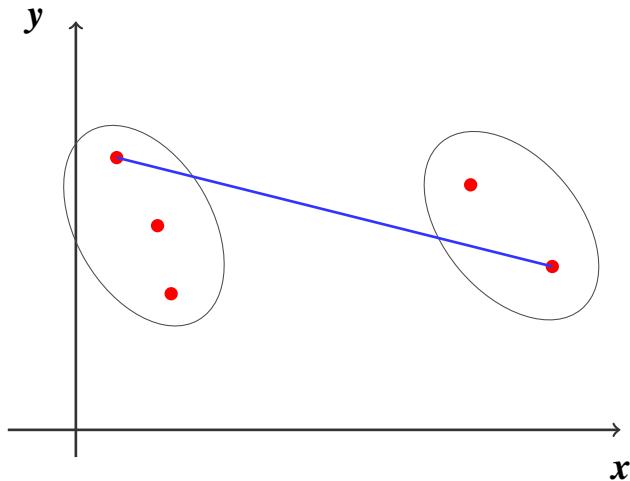
グループ間距離の再計算のイメージ



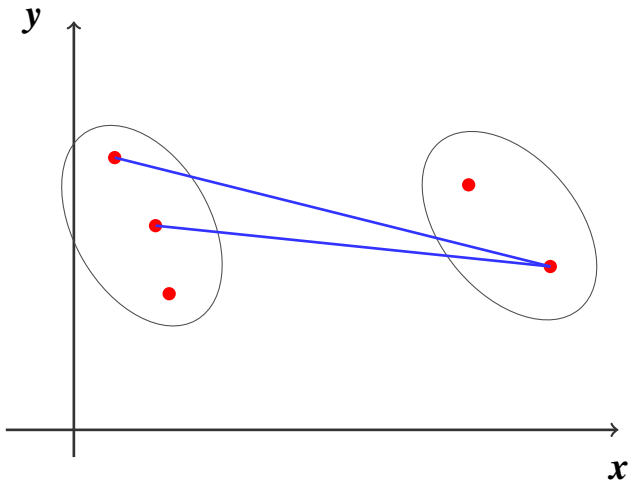
グループ間距離の再計算のイメージ



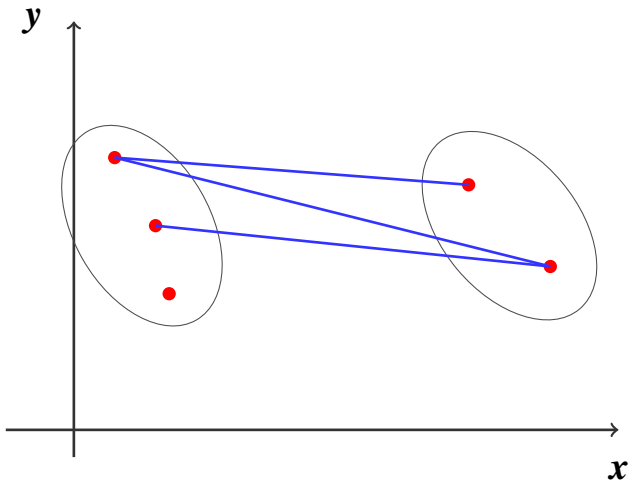
グループ間距離の再計算のイメージ



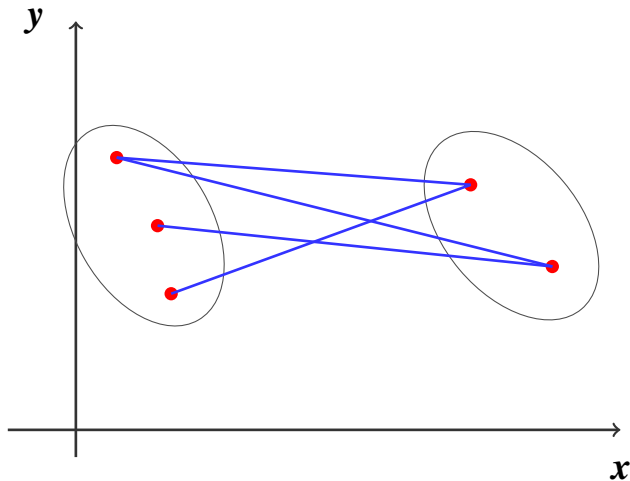
グループ間距離の再計算のイメージ



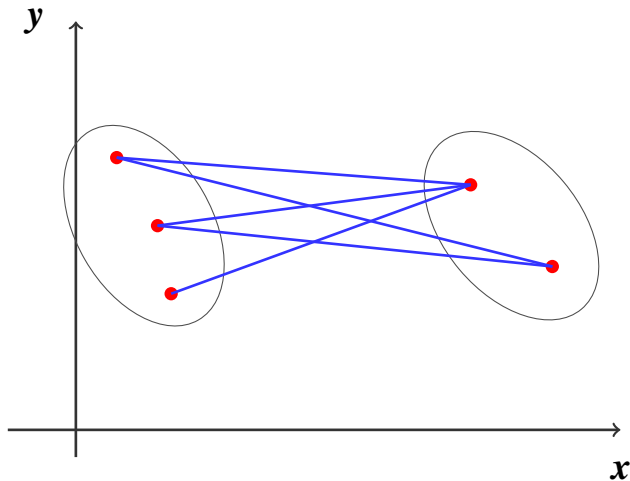
グループ間距離の再計算のイメージ



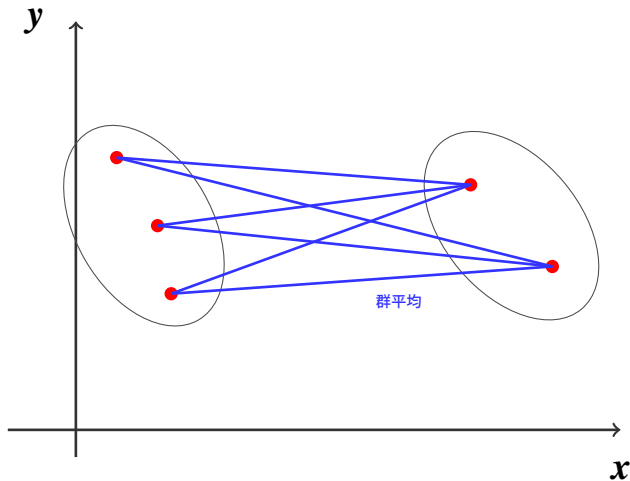
グループ間距離の再計算のイメージ



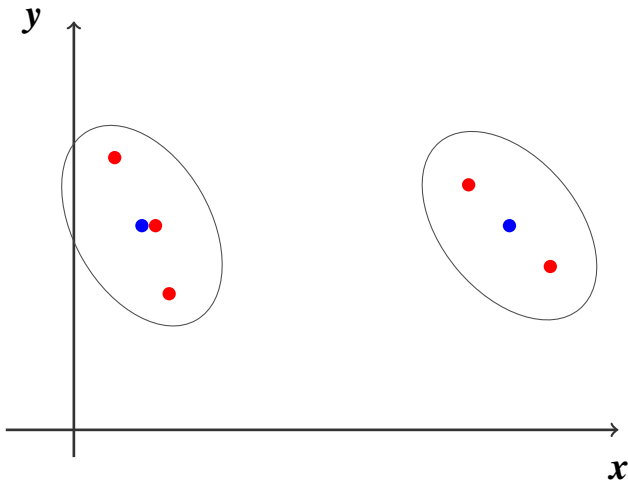
グループ間距離の再計算のイメージ



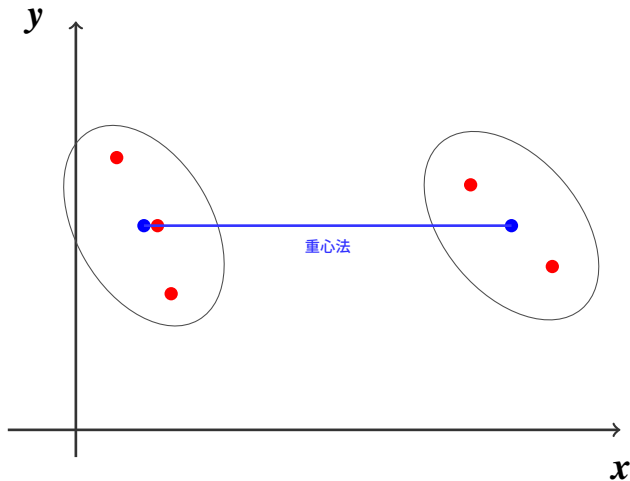
グループ間距離の再計算のイメージ



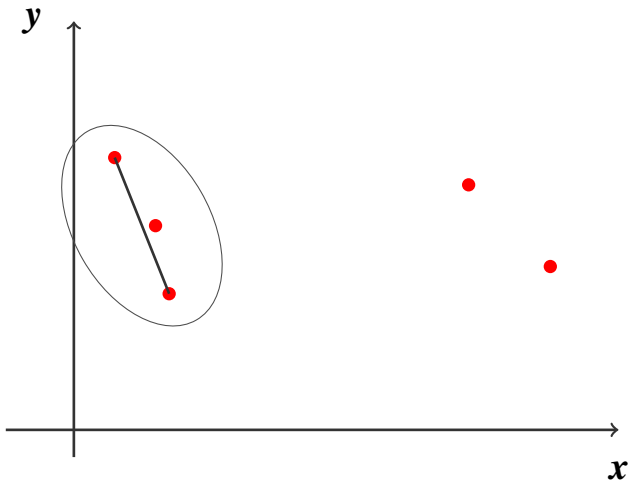
距離の再計算のイメージ



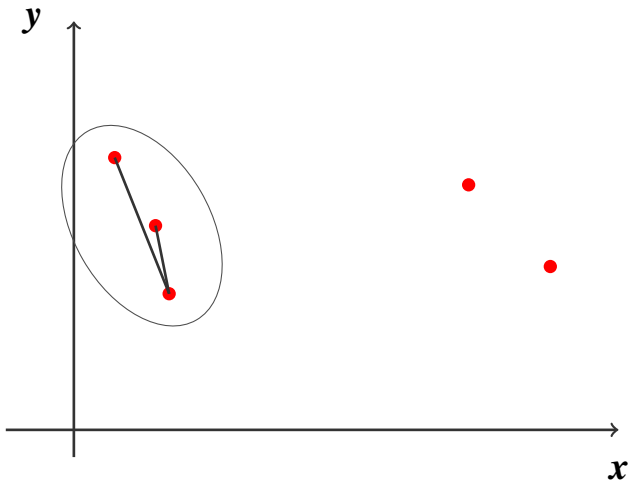
距離の再計算のイメージ



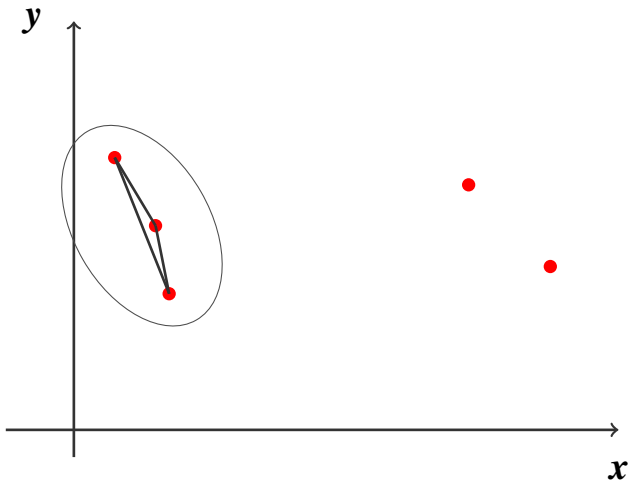
距離の再計算のイメージ



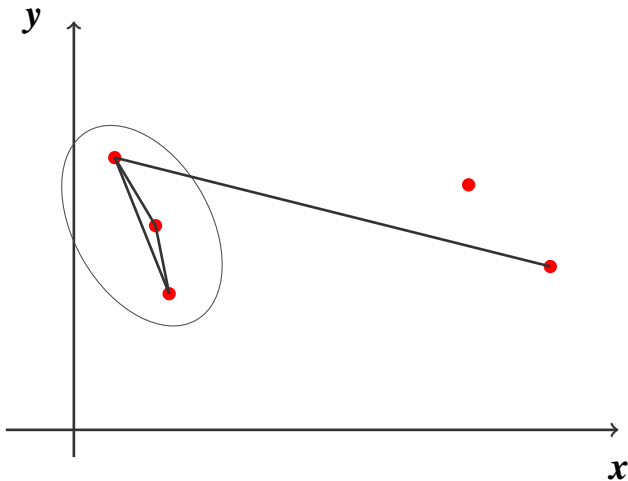
距離の再計算のイメージ



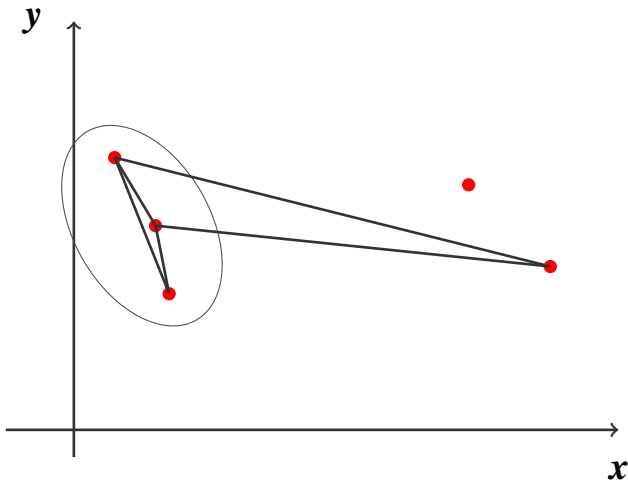
距離の再計算のイメージ



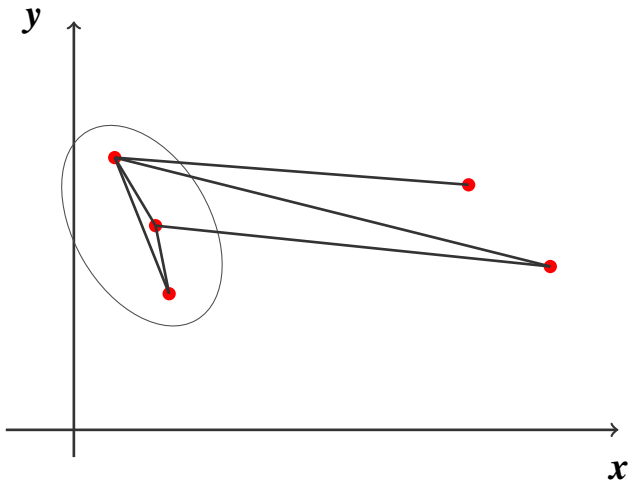
距離の再計算のイメージ



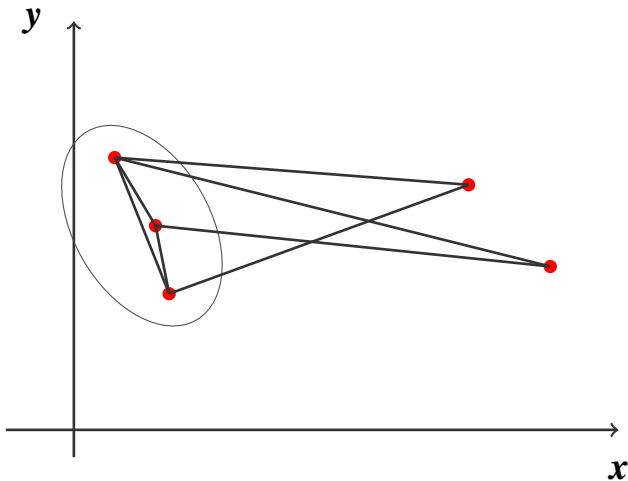
距離の再計算のイメージ



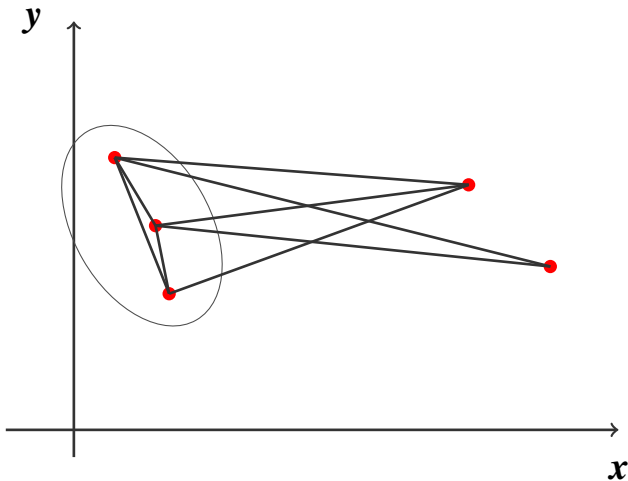
距離の再計算のイメージ



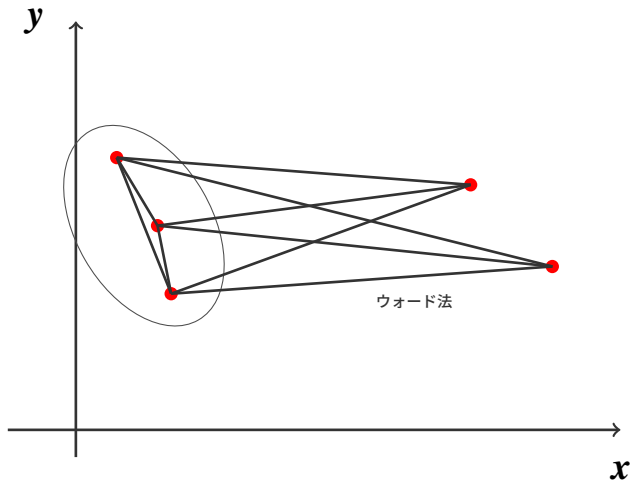
距離の再計算のイメージ



距離の再計算のイメージ



距離の再計算のイメージ



距離の再計算法

ワード法

- クラスターが ようにグループの中の距離がどれだけ増えるかを計算し、それが となる点を選ぶ
- が形成している集団をみつけない場合に用いる
 - 最短距離法の場合、クラスターの内側に他のクラスターと近い点があればそれが採用されクラスターを結合していき鎖状の樹形図が作られやすいので、それを避けるということ

距離の再計算法

ワード法

- クラスターが **巨大にならない** ようにグループの中の距離がどれだけ増えるかを計算し、それが となる点を選ぶ
- が形成している集団をみつけない場合に用いる
 - 最短距離法の場合、クラスターの内側に他のクラスターと近い点があればそれが採用されクラスターを結合していき鎖状の樹形図が作られやすいので、それを避けるということ

距離の再計算法

ワード法

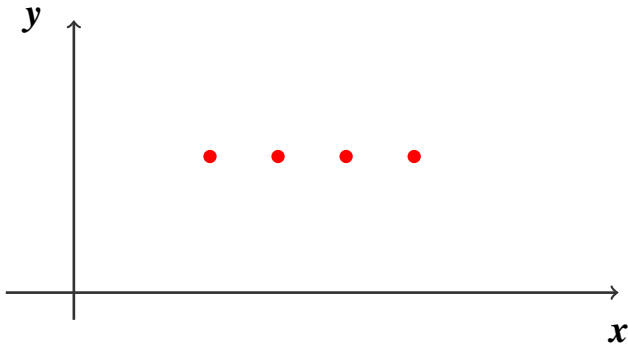
- クラスターが **巨大にならない** ようにグループの中の距離がどれだけ増えるかを計算し、それが **一番最小** となる点を選ぶ
- が形成している集団をみつけない場合に用いる
 - 最短距離法の場合、クラスターの内側に他のクラスターと近い点があればそれが採用されクラスターを結合していき鎖状の樹形図が作られやすいので、それを避けるということ

距離の再計算法

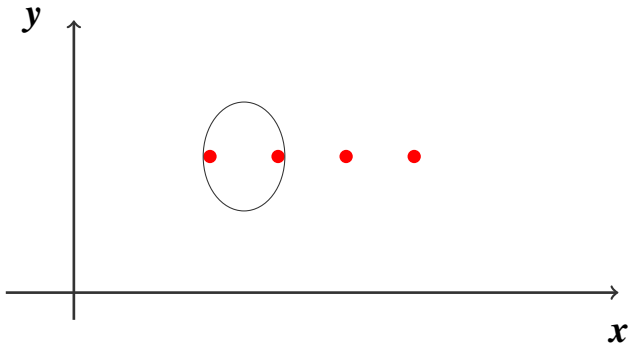
ワード法

- クラスターが **巨大にならない** ようにグループの中の距離がどれだけ増えるかを計算し、それが **一番最小** となる点を選ぶ
- **近いもの同士** が形成している集団をみつけない場合に用いる
 - 最短距離法の場合、クラスターの内側に他のクラスターと近い点があればそれが採用されクラスターを結合していき鎖状の樹形図が作られやすいので、それを避けるということ

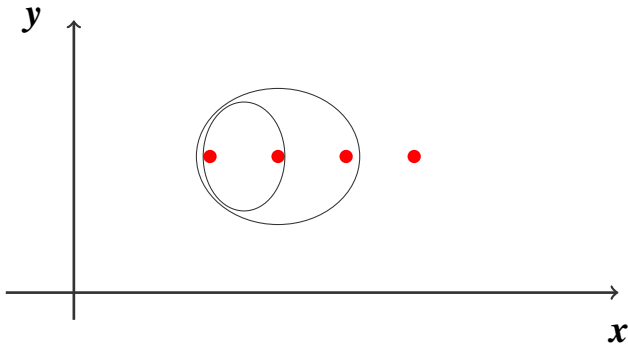
最短距離法のイメージ



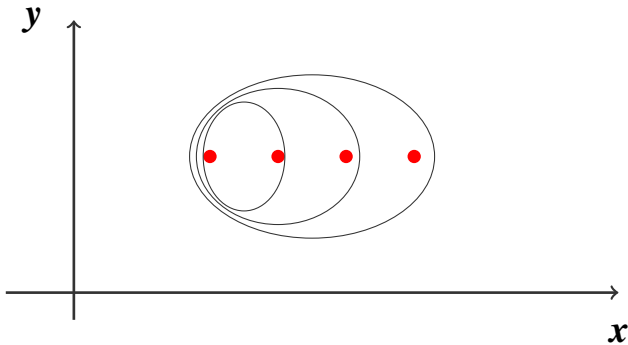
最短距離法のイメージ



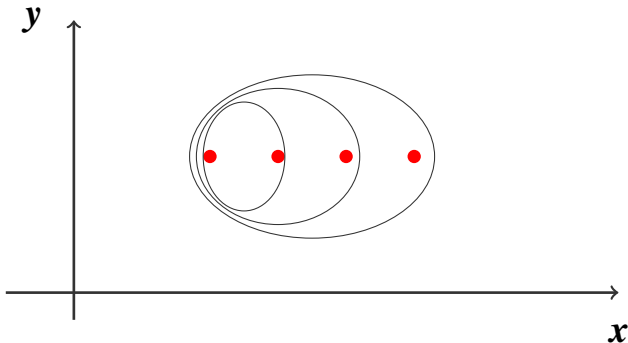
最短距離法のイメージ



最短距離法のイメージ



最短距離法のイメージ



- 極端な場合、かたよったクラスターとなる

クラスター分析【練習】

1 以下のデータを I:\rdata\ cl01.csv に保存する

```
, "大阪", "京橋", "鶴橋", "天王", "新今", "西九"  
"大阪", 0, 7, 16, 22, 18, 7  
"京橋", 7, 0, 9, 16, 20, 14  
"鶴橋", 16, 9, 0, 7, 11, 22  
"天王寺", 22, 16, 7, 0, 4, 15  
"新今宮", 18, 20, 11, 4, 0, 11  
"西九条", 7, 14, 22, 15, 11, 0
```

前週の mds01.csv と同じものなので、それを使っても可。その場合は次頁の cl01.csv のところを mds01.csv とする。

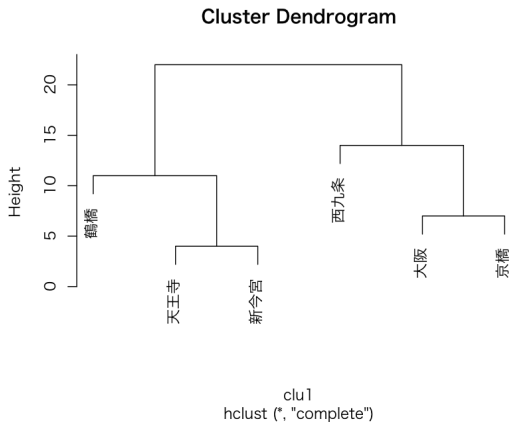
クラスター分析【練習】

2 R を立ち上げ、以下を実行する

```
clu0 <- read.csv('I:/rdata/cl01.csv',
,header=T,row.names=1) # 本当は1行
clu1 <- as.dist(clu0) # 下の準備
clu2 <- hclust(clu1) # 階層的クラスター
plot(clu2) # ↑オプションなしは最長距離法
clu3 <- hclust(clu1,method="ward.D")
# ウォード法を用いる場合 (.D が必要)
plot(clu3)
```

クラスター分析【練習】

☆ 出力結果



クラスター分析【練習】

☆ hclust 関数のオプションについて
(method="..." で指定)

方法	引数
最短距離法	
最長距離法	
群平均法	
重心法	
ワード法	ward.D (ward は今のバージョンでは無効)

クラスター分析【練習】

☆ hclust 関数のオプションについて
(method="..." で指定)

方法	引数
最短距離法	single
最長距離法	
群平均法	
重心法	
ワード法	ward.D (ward は今のバージョンでは無効)

クラスター分析【練習】

☆ hclust 関数のオプションについて
(method="..." で指定)

方法	引数
最短距離法	single
最長距離法	complete
群平均法	
重心法	
ワード法	ward.D (ward は今のバージョンでは無効)

クラスター分析【練習】

☆ hclust 関数のオプションについて
(method="..." で指定)

方法	引数
最短距離法	single
最長距離法	complete
群平均法	average
重心法	
ワード法	ward.D (ward は今のバージョンでは無効)

クラスター分析【練習】

☆ hclust 関数のオプションについて
(method="..." で指定)

方法	引数
最短距離法	single
最長距離法	complete
群平均法	average
重心法	centroid
ワード法	ward.D (ward は今のバージョンでは無効)

非階層的クラスター分析

- 代表例：

- データを k 個のクラスターに分ける方法

非階層的クラスター分析

- 代表例： **k-means (K 平均) 法**
- データを k 個のクラスターに分ける方法

k-means 法

- 1 まず k 個の点を に選び、
 と考える（グループに分けるやり方もある）
- 2 残りの $N - k$ 個の点に対して k 個の代表点との距離を計算し、 グループに属することにする。これにより全ての点が、取り敢えず k 個のグループに分かれる
- 3 これら k 個のグループごとに、それぞれの座標から の点の座標を計算し、その点を新たにグループの代表点とする

k-means 法

- 1 まず k 個の点を **ランダム** に選び、
 と考える（グループに分けるやり方もある）
- 2 残りの $N - k$ 個の点に対して k 個の代表点との距離を計算し、 グループに属することにする。これにより全ての点が、取り敢えず k 個のグループに分かれる
- 3 これら k 個のグループごとに、それぞれの座標から の点の座標を計算し、その点を新たにグループの代表点とする

k-means 法

- 1 まず k 個の点を **ランダム** に選び、
グループの中心 と考える（グループに分けるやり方もある）
- 2 残りの $N - k$ 個の点に対して k 個の代表点との距離を計算し、 グループに属することにする。これにより全ての点が、取り敢えず k 個のグループに分かれる
- 3 これら k 個のグループごとに、それぞれの座標から の点の座標を計算し、その点を新たにグループの代表点とする

k-means 法

- 1 まず k 個の点を **ランダム** に選び、
グループの中心 と考える（グループに分けるやり方もある）
- 2 残りの $N - k$ 個の点に対して k 個の代表点との距離を計算し、**一番近い** グループに属することにする。これにより全ての点が、取り敢えず k 個のグループに分かれる
- 3 これら k 個のグループごとに、それぞれの座標から の点の座標を計算し、その点を新たにグループの代表点とする

k-means 法

- 1 まず k 個の点を **ランダム** に選び、
グループの中心 と考える（グループに分けるやり方もある）
- 2 残りの $N - k$ 個の点に対して k 個の代表点との距離を計算し、**一番近い** グループに属することにする。これにより全ての点が、取り敢えず k 個のグループに分かれる
- 3 これら k 個のグループごとに、それぞれの座標から **重心** の点の座標を計算し、その点を新たにグループの代表点とする

k-means 法（続き）

- 4 全部の点に関してその代表の点との距離を
、一番近いグループに属するようにグループを入れかえる
- 5 上の手順を繰り返す
- 6 グループに まで繰り返す（終了条件は回数で切るなど、問題に応じて使い分ける）

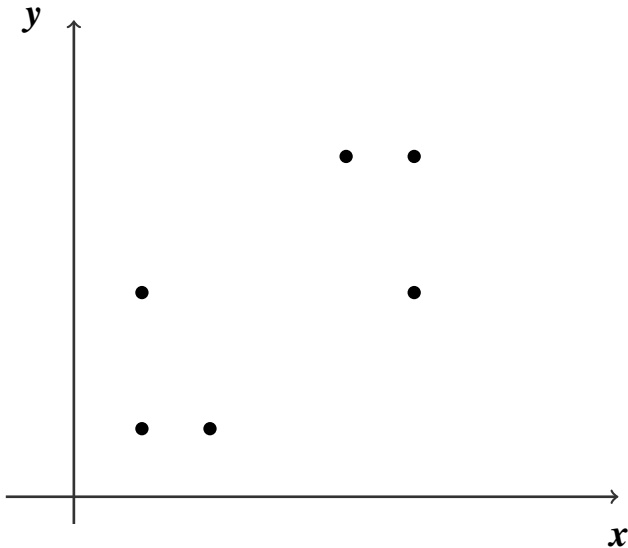
k-means 法（続き）

- 4 全部の点に関してその代表の点との距離を
計算し直し、一番近いグループに属するようにグループを入れかえる
- 5 上の手順を繰り返す
- 6 グループに まで繰り返す（終了条件は回数で切るなど、問題に応じて使い分ける）

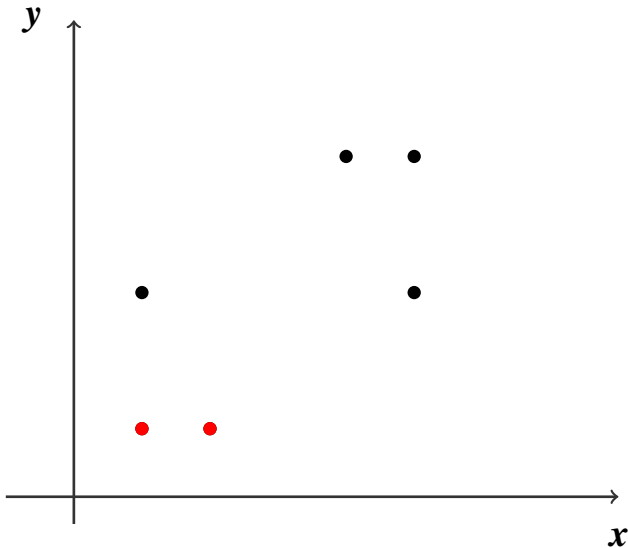
k-means 法（続き）

- 4 全部の点に関してその代表の点との距離を
計算し直し、一番近いグループに属するようにグループを入れかえる
- 5 上の手順を繰り返す
- 6 グループに **変化がなくなる** まで繰り返す（終了条件は回数で切るなど、問題に応じて使い分ける）

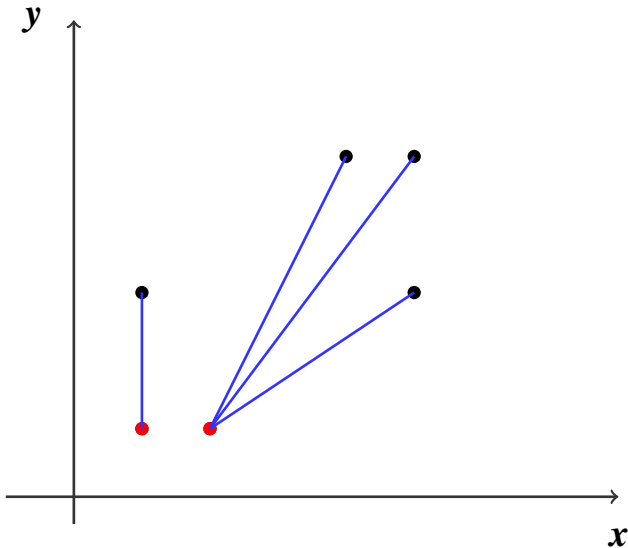
k-means 法のイメージ



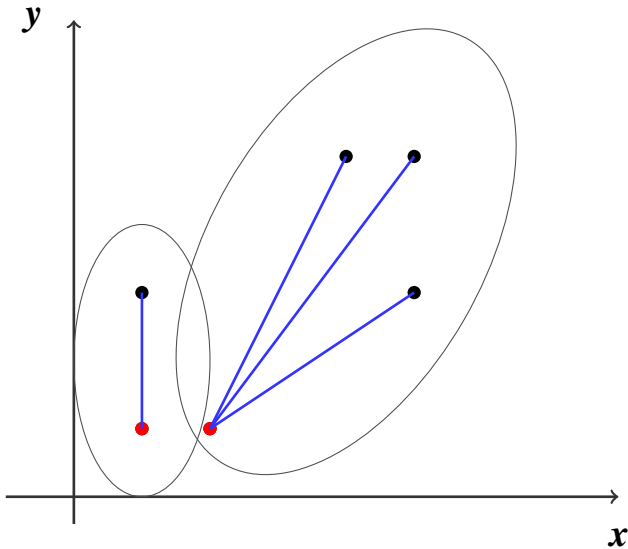
k-means 法のイメージ



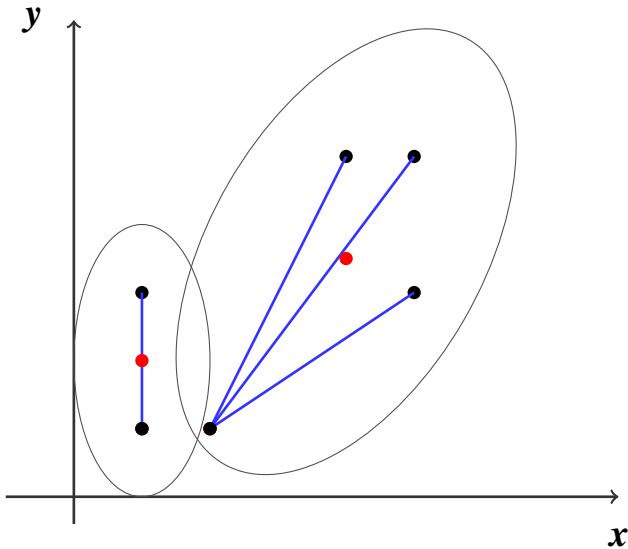
k-means 法のイメージ



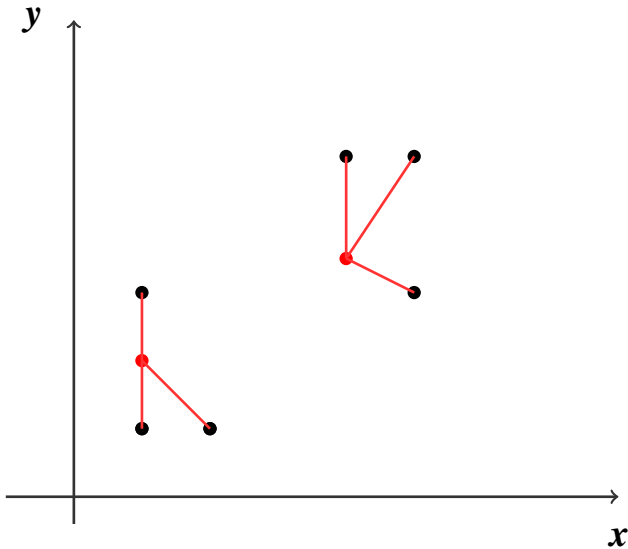
k-means 法のイメージ



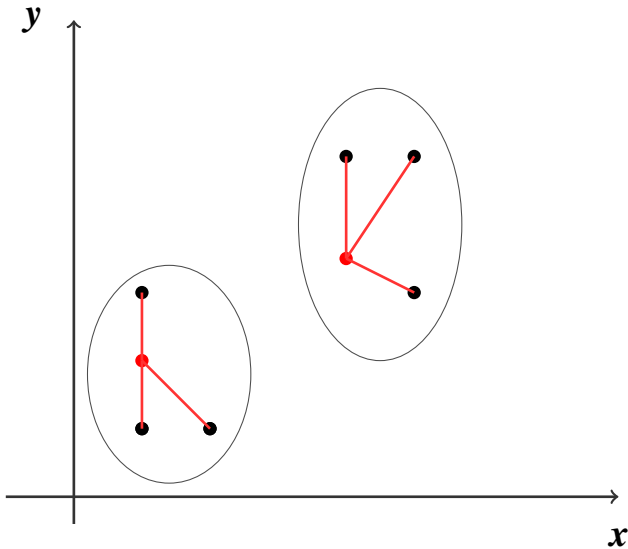
k-means 法のイメージ



k-means 法のイメージ



k-means 法のイメージ



k-means 法（具体例）

- 1 以下の6点があり、距離はユークリッド距離とする

点	(x,y)
C1	(1,1)
C2	(2,1)
C3	(1,3)
C4	(4,5)
C5	(5,5)
C6	(5,3)

k-means 法（具体例）

- 2 C1,C2 が初期点として選ばれたとする
- 3 C1 に近い点として C3 が、C2 に近い点として C4,C5,C6 が当てはまる
- 4 $\rightarrow \{C1,C3\}$ と $\{C2,C4,C5,C6\}$ の2つのクラスターに分かれる
- 5 それぞれのクラスターの中心の点は とする

k-means 法（具体例）

- 2 C1,C2 が初期点として選ばれたとする
- 3 C1 に近い点として C3 が、C2 に近い点として C4,C5,C6 が当てはまる
- 4 $\rightarrow \{C1,C3\}$ と $\{C2,C4,C5,C6\}$ の2つのクラスターに分かれる
- 5 それぞれのクラスターの中心の点は **重心** とする

k-means 法（具体例）

- 6 C1,C3 の重心は $(\frac{1+1}{2}, \frac{1+3}{2}) = (1, 2)$, C2,C4,C5,C6 の重心は $(\frac{2+4+5+5}{4}, \frac{1+5+5+3}{4}) = (4, 3.5)$ となる
- 7 もう一度この点から近い点と遠い点でグループ分けをすると、 $\{C1, C2, C3\}$ と $\{C4, C5, C6\}$ からなるクラスターに分かれる
- 8 これに基づいて再度計算しても点の入れ替わりはないので終了となる

k-means 法（具体的計算）

- 1 それぞれの距離の 2 乗を表にし、どの点に近いかを表にする

点	C1	C2	
C1	0	2	-
C2	2	0	-
C3	4	5	C1
C4	25	20	C2
C5	32	25	C2
C6	20	13	C2

k-means 法（具体的計算）

2 (ステップ2)

点	P1(1,2)	P2(4,3.5)	
C1	1	15.25	P1
C2	2	10.25	P1
C3	1	9.25	P1
C4	18	2.25	P2
C5	25	3.25	P2
C6	17	1.25	P2

3 それぞれの中心の座標は、 $Q1=(\frac{4}{3}, \frac{5}{3})$, $Q2=(\frac{14}{3}, \frac{13}{3})$

k-means 法（具体的計算）

4 (ステップ3)

点	Q1(1.333,1.667)	Q2(4.667,4.333)	
C1	0.566	24.556	Q1
C2	0.889	18.222	Q1
C3	1.889	15.222	Q1
C4	18.222	0.889	Q2
C5	24.556	0.556	Q2
C6	15.222	1.889	Q2

5 クラスターの点の入れ替えはおこらないので、終了

k-means 【練習】

- 1 以下のデータを R: ¥ rdata ¥ km01.csv に保存する

name, x, y

C1, 1, 1

C2, 2, 1

C3, 1, 3

C4, 4, 5

C5, 5, 5

C6, 5, 3

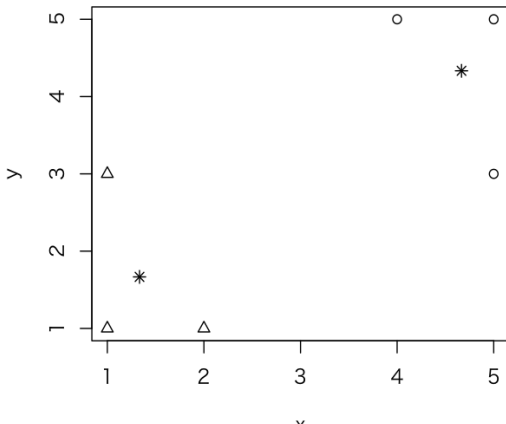
クラスター分析【練習】

2 R を立ち上げ、以下を実行する

```
km1 <- read.csv('I:/rdata/km01.csv',  
  ,header=T,row.names=1) # 本当は1行  
km2 <- kmeans(km1,2) # 2 クラスター  
plot(km1,pch=km2$cluster) # プロット  
points(km2$centers,pch=8) # ポイント
```

k-means 【練習】

☆ 出力結果



おまけ：クラスター数

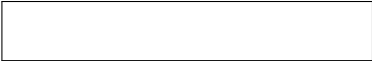
- k-means 法の場合、事前にクラスター数 k を決めて、それに従ってクラスタリングを行なう
- その k が、適切なのか？の検証が必要
- →クラスター数の を用いる
 -
 -
 -

おまけ：クラスター数

- k-means 法の場合、事前にクラスター数 k を決めて、それに従ってクラスタリングを行なう
- その k が、適切なのか？の検証が必要
- →クラスター数の **評価方法** を用いる

- 
-
-

おまけ：クラスター数

- k-means 法の場合、事前にクラスター数 k を決めて、それに従ってクラスタリングを行なう
- その k が、適切なのか？の検証が必要
- →クラスター数の **評価方法** を用いる
 - **エルボー法**
 - 
 - 

おまけ：クラスター数

- k-means 法の場合、事前にクラスター数 k を決めて、それに従ってクラスタリングを行なう
- その k が、適切なのか？の検証が必要
- →クラスター数の **評価方法** を用いる
 - **エルボー法**
 - **シルエット分析法**
 -

おまけ：クラスター数

- k-means 法の場合、事前にクラスター数 k を決めて、それに従ってクラスタリングを行なう
- その k が、適切なのか？の検証が必要
- →クラスター数の **評価方法** を用いる
 - **エルボー法**
 - **シルエット分析法**
 - **ギャップ統計法**

おまけ：エルボー法

- クラスター内変動（距離、誤差）の平方和を求めると、 k が増加するほど、クラスター内変動は下がり続ける
- k とその変動をプロットすると変動が急降下した後、変動が ポイントがある＝肘（エルボー）
- 肘に当たる k を、最適なクラスター数とする
- 課題：肘が明確に描画されるプロットはあまり

おまけ：エルボー法

- クラスター内変動（距離、誤差）の平方和を求めると、 k が増加するほど、クラスター内変動は下がり続ける
- k とその変動をプロットすると変動が急降下した後、変動が **安定する** ポイントがある＝肘（エルボー）
- 肘に当たる k を、最適なクラスター数とする
- 課題：肘が明確に描画されるプロットはあまり



おまけ：エルボー法

- クラスター内変動（距離、誤差）の平方和を求めると、 k が増加するほど、クラスター内変動は下がり続ける
- k とその変動をプロットすると変動が急降下した後、変動が **安定する** ポイントがある＝肘（エルボー）
- 肘に当たる k を、最適なクラスター数とする
- 課題：肘が明確に描画されるプロットはあまり **現れない**

おまけ：シルエット分析法

- $$\text{スコア} = \frac{\text{乖離度} - \text{凝集度}}{\max(\text{乖離度}, \text{凝集度})}$$
 - 凝集度：クラスタ内の全データ同士の平均距離
 - 乖離度：隣接するクラスタ同士の全データの平均距離
- スコアで判定
 - $1 \sim 0$: 分離性能が良い
 - ≈ 0 : クラスタリングの分離性能が悪い
 - $-1 \sim 0$: 間違ったクラスタに所属している可能性がある。
- 平均シルエット・スコアを計算し最も
な k を最適なクラスター数とする

おまけ：シルエット分析法

- $$\text{スコア} = \frac{\text{乖離度} - \text{凝集度}}{\max(\text{乖離度}, \text{凝集度})}$$
 - 凝集度：クラスタ内の全データ同士の平均距離
 - 乖離度：隣接するクラスタ同士の全データの平均距離
- スコアで判定
 - $1 \sim 0$ ：分離性能が良い
 - ≈ 0 ：クラスタリングの分離性能が悪い
 - $-1 \sim 0$ ：間違ったクラスタに所属している可能性がある。
- 平均シルエット・スコアを計算し最も
スコアが良好な k を最適なクラスター数とする

おまけ：ギャップ統計法

- 対象データと、その同範囲で一様分布（クラスター数 1）する架空のデータをクラスタリングし、それぞれのクラスター同士の を比較する
- ギャップを最大化する最小の k を、最適なクラスター数とする
- 利点：プロットの見た目に関わらず に最適値が出力される

おまけ：ギャップ統計法

- 対象データと、その同範囲で一様分布（クラスター数 1）する架空のデータをクラスタリングし、それぞれのクラスター同士の **変動（ギャップ）** を比較する
- ギャップを最大化する最小の k を、最適なクラスター数とする
- 利点：プロットの見た目に関わらず に最適値が出力される

おまけ：ギャップ統計法

- 対象データと、その同範囲で一様分布（クラスター数 1）する架空のデータをクラスタリングし、それぞれのクラスター同士の **変動（ギャップ）** を比較する
- ギャップを最大化する最小の k を、最適なクラスター数とする
- 利点：プロットの見た目に関わらず **機械的** に最適値が出力される

おしまい

質問があれば、お気軽に！