

# 11. モデル推定

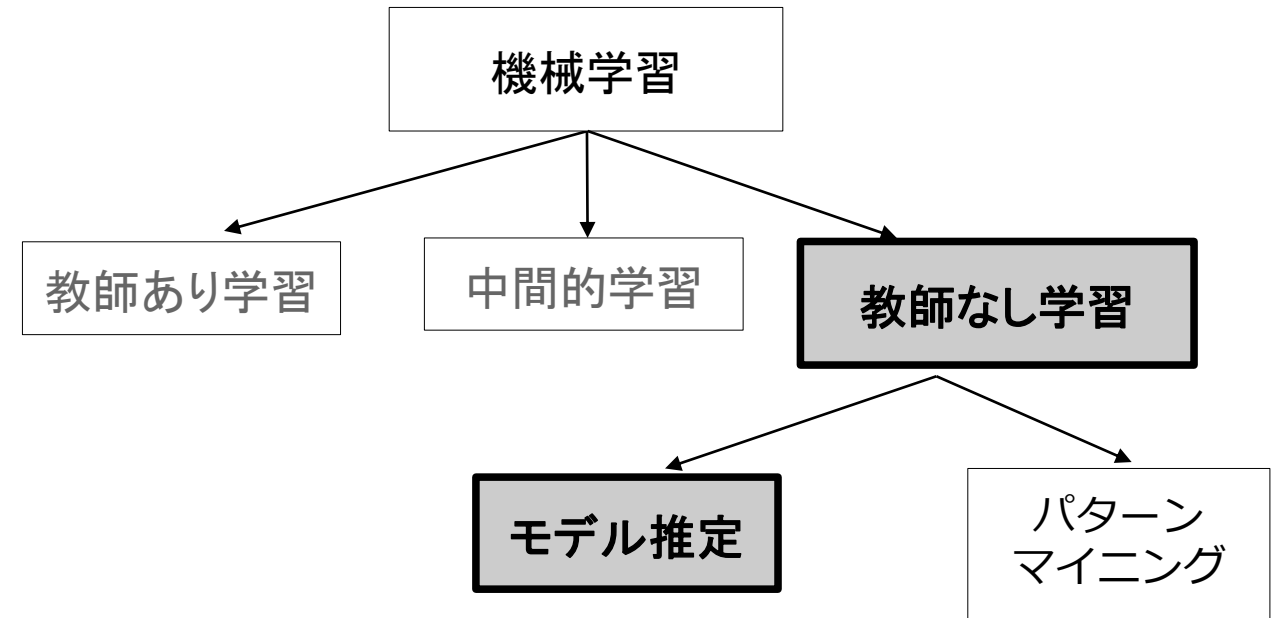
- 本章の説明手順
  - ◆ 教師なし、モデル推定の問題設定
  - ◆ ボトムアップにデータをまとめてゆく手法
  - ◆ トップダウンにデータの分割を行い最適化してゆく手法
  - ◆ まとまりの確率分布を推定する手法

## 11.1 数値特徴に対する「教師なし・モデル推定」問題の定義

- 問題設定

- ◆ 教師なし学習

- 数値ベクトル → クラスモデル
    - データ全体を説明するモデルを見つける
    - 応用例
      - ✓ 顧客セグメンテーション
      - ✓ 異常検出



## 11.1 数値特徴に対する「教師なし・モデル推定」問題の定義

- データセット（教師なし）

- ◆ （密な）数値ベクトル

$$\{\mathbf{x}_i\} \quad i = 1, \dots, N$$

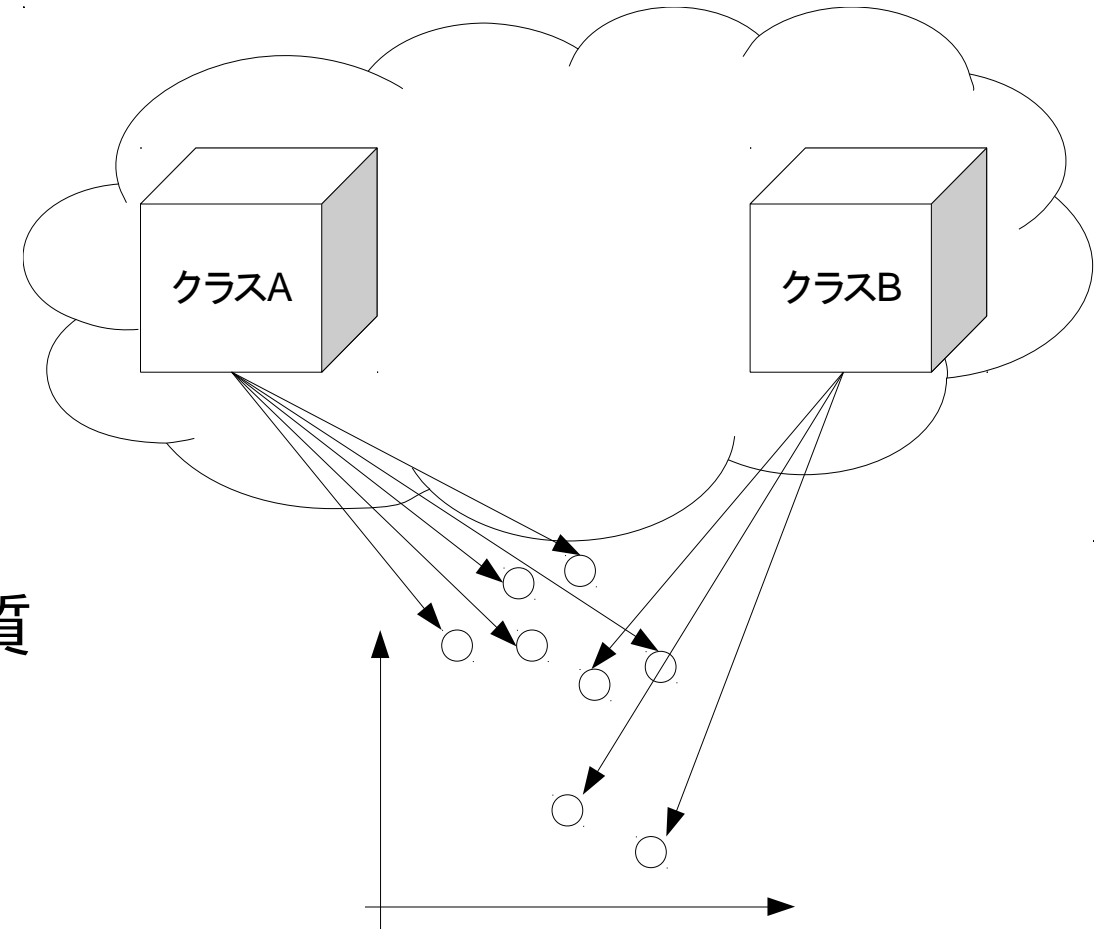
- モデル推定とは

- ◆ クラスタリング

- 個々のデータを生じさせた共通の性質をもつクラスを見つける

- ◆ 確率密度推定

- そのクラスの統計的性質を推定する



# 11.2 クラスタリング

- クラスタリングとは
  - ◆ 「共通の性質をもつクラス」 = 「特徴空間上で近い値をもつデータの集まり」と考え、データのまとまりを見つける
  - ◆ 「まとまり」とは
    - 内的結合（同じ集合内のデータ間の距離は小さく）  
と  
外的分離（異なる集合間の距離は大きく）  
が同時に満たされる部分集合

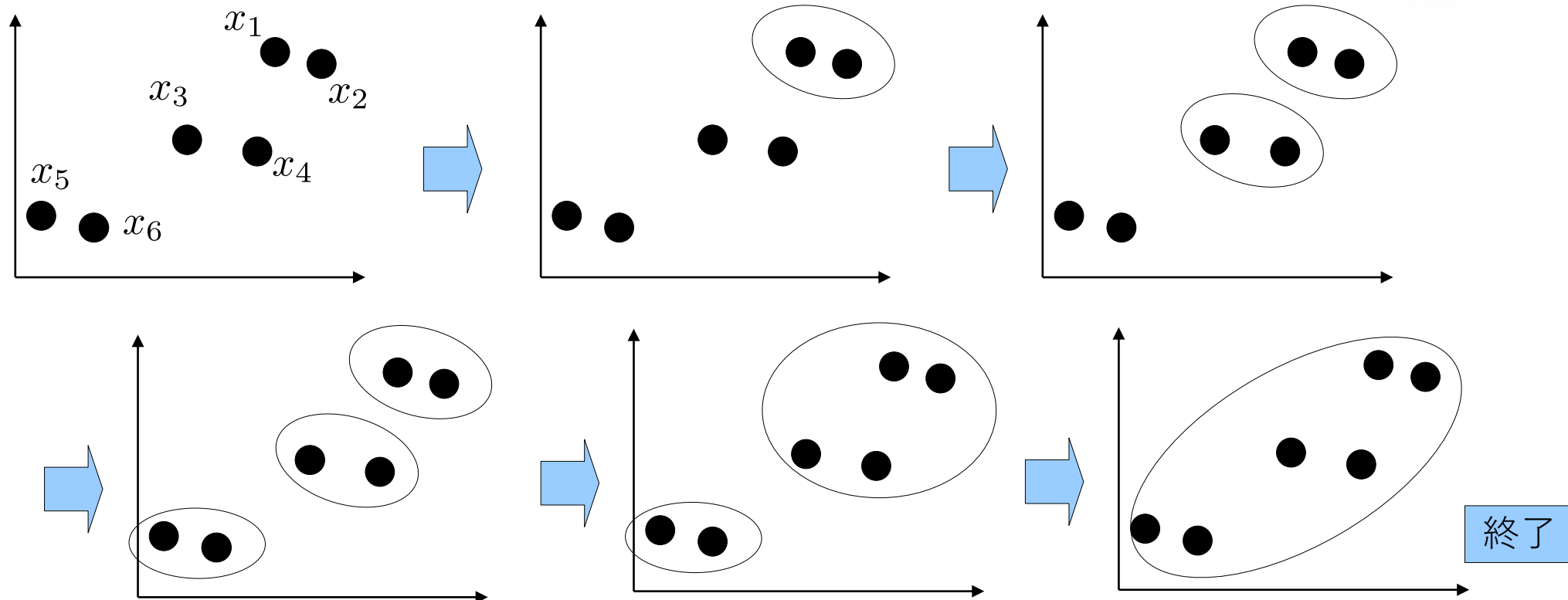
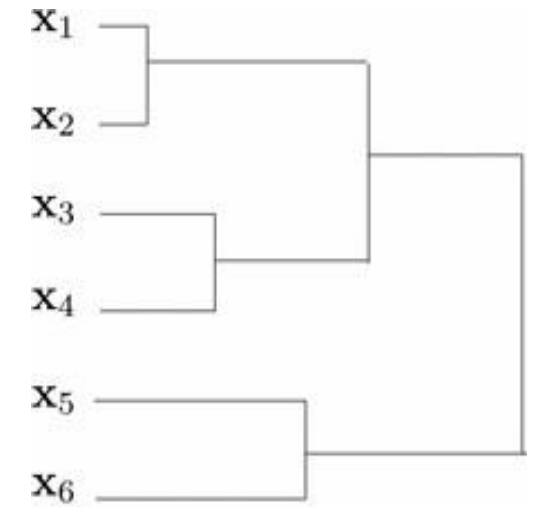
## 11.2 クラスタリング

- クラスタリング手法の分類
  - ◆ 階層的手法
    - ボトムアップ的にデータをまとめてゆく
  - ◆ 分割最適化手法
    - トップダウン的にデータ集合を分割し、最適化してゆく

# 11.2.1 階層的クラスタリング

- 階層的クラスタリングとは

- 1データ1クラスタからスタート
- 最も近接するクラスタを求めて、1つにまとめる
- 2.を繰り返し、全データが1クラスタになれば終了



# 11.2.1 階層的クラスタリング

---

Algorithm 11.1 階層的クラスタリング

---

入力: 正解なしデータ  $D$

出力: クラスタリング結果の木構造

/\* 学習データそれぞれをクラスタの要素としたクラスタ集合  $C$  を作成 \*/

$C \leftarrow \{c_1, c_2, \dots, c_N\}$

**while**  $|C| > 1$  **do**

/\* もっとも似ているクラスタ対  $\{c_m, c_n\}$  を見つける \*/

$(c_m, c_n) \leftarrow \underset{c_i, c_j \in C}{\operatorname{argmax}} \operatorname{sim}(c_i, c_j)$

$\{c_m, c_n\}$  を融合

**end while**

---

# 11.2.1 階層的クラスタリング

- 類似度simの定義（正確にはこれらの反数）
  - ◆ 単連結法(single)
    - 類似度：最も近いデータ対の距離
    - 傾向：クラスタが一方向に伸びやすくなる
  - ◆ 完全連結法(complete)
    - 類似度：最も遠いデータ対の距離
    - 傾向：クラスタが一方向に伸びるのを避ける
  - ◆ 重心法(average)
    - 類似度：クラスタの平均ベクトル間の距離
    - 傾向：単連結と完全連結の中間的な形
  - ◆ Ward法(ward)
    - 類似度：融合前後の「平均ベクトルとデータの距離の二乗和」の差
    - 傾向：極端な形になりにくく、よく用いられる基準



## 11.2.2 分割最適化クラスタリング

- 分割最適化クラスタリングとは
  - ◆ データ分割の良さを評価する関数を定め、その評価関数の値を最適化することを目的とする
  - ◆ ただし、すべての可能な分割に対して評価値を求めることは、データ数 $N$ が大きくなると不可能
    - 例：2分割で $2^N$ 通り
  - ◆ 探索によって、準最適解を求める

# 11.2.2 分割最適化クラスタリング

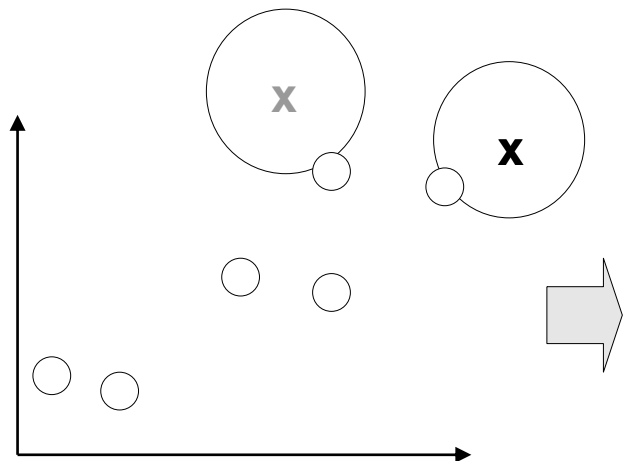
- k-meansアルゴリズム

評価関数：データとクラスタ中心との距離の和

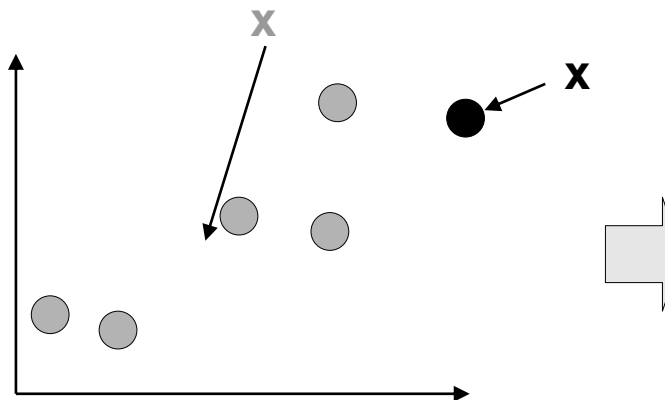
1. 分割数 $k$ を予め与える

2. 乱数で $k$ 個のクラスタ中心を設定し、逐次更新

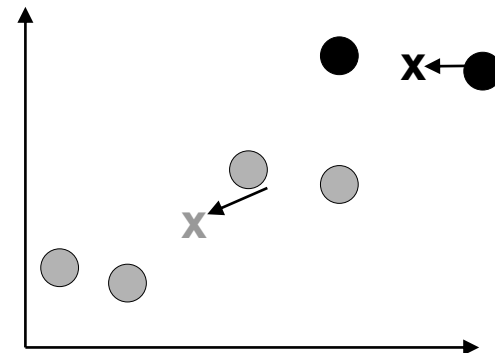
① 初期値として乱数でクラスタ中心を配置



② 各データを、もっとも近いクラスタに配属  
③ 所属しているデータからクラスタ中心を再計算



④ ②, ③の処理をクラスタ中心が動かなくなるまで繰り返す



# 11.2.2 分割最適化クラスタリング

---

Algorithm 11.2 k-means アルゴリズム

---

入力: 正解なしデータ  $D$

出力: クラスタ中心  $\mu_j$  ( $j = 1, \dots, k$ )

入力空間上に  $k$  個の点をランダムに設定し, それらをクラスタ中心  $\mu_j$  とする

repeat

  for all  $x_i \in D$  do

    各クラスタ中心  $\mu_j$  との距離を計算し, もっとも近いクラスタに割り当てる

  end for

/\* 各クラスタについて, 以下の式で中心の位置を更新

  ( $N_j$  はクラスタ  $j$  のデータ数) \*/

$$\mu_j \leftarrow \frac{1}{N_j} \sum_{x_i \in \text{クラスタ}_j} x_i \quad (j = 1, \dots, k)$$

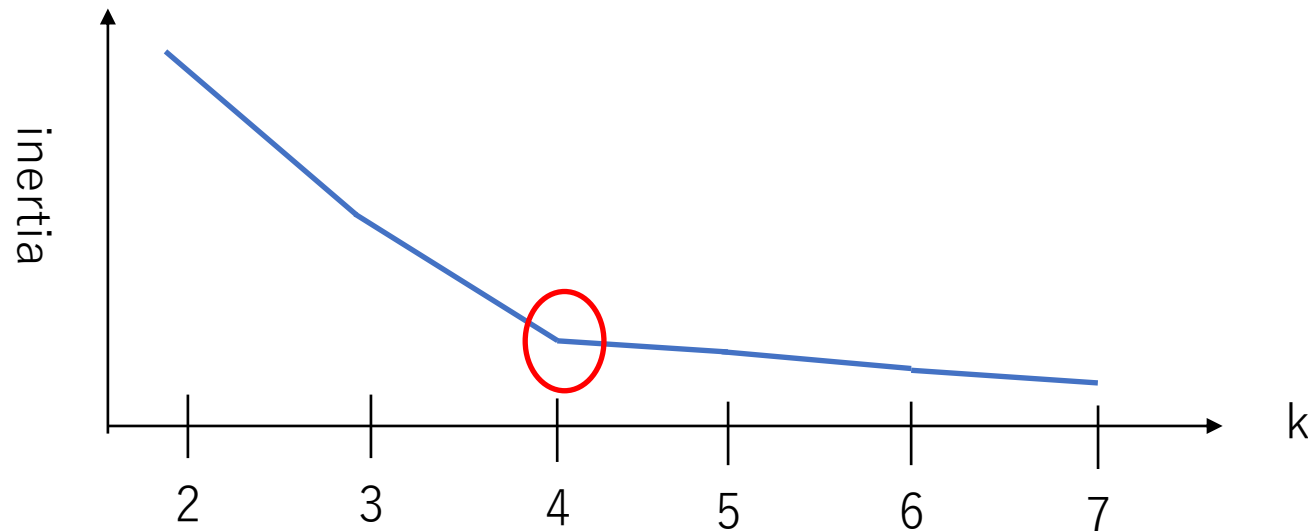
until クラスタ中心  $\mu_j$  が変化しない

return  $\mu_j$  ( $j = 1, \dots, k$ )

---

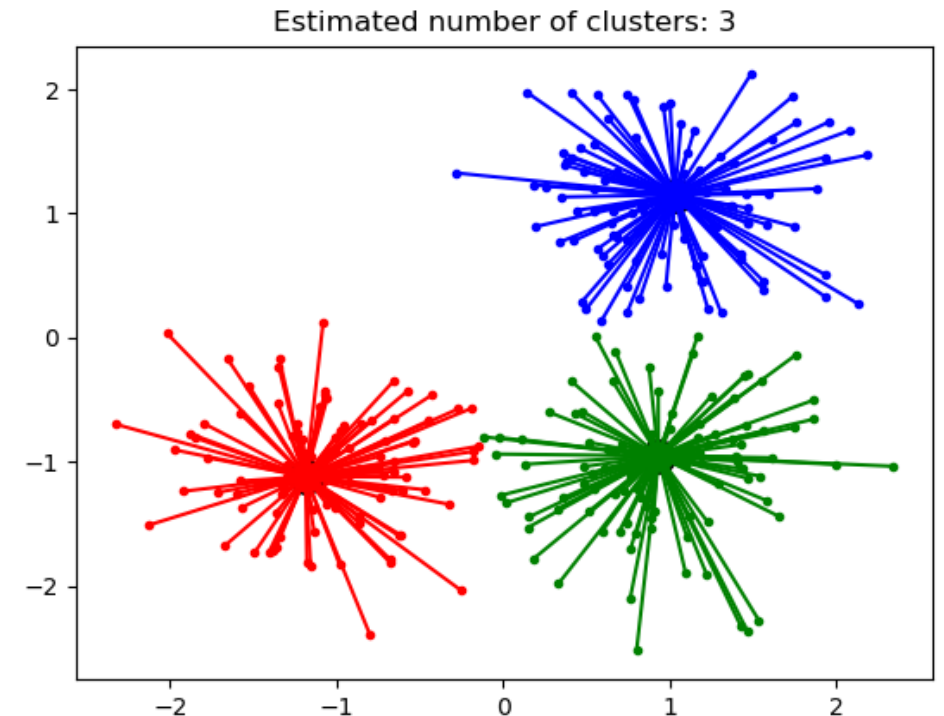
## 11.2.2 分割最適化クラスタリング

- k-means法の問題点 1
  - ◆ 分割数 $k$ を予め決めなければならない
- 解決法
  - ◆ エルボーメソッド
    - データとクラスタ中心との平均二乗距離 (inertia) の減り方が鈍るところを見つける



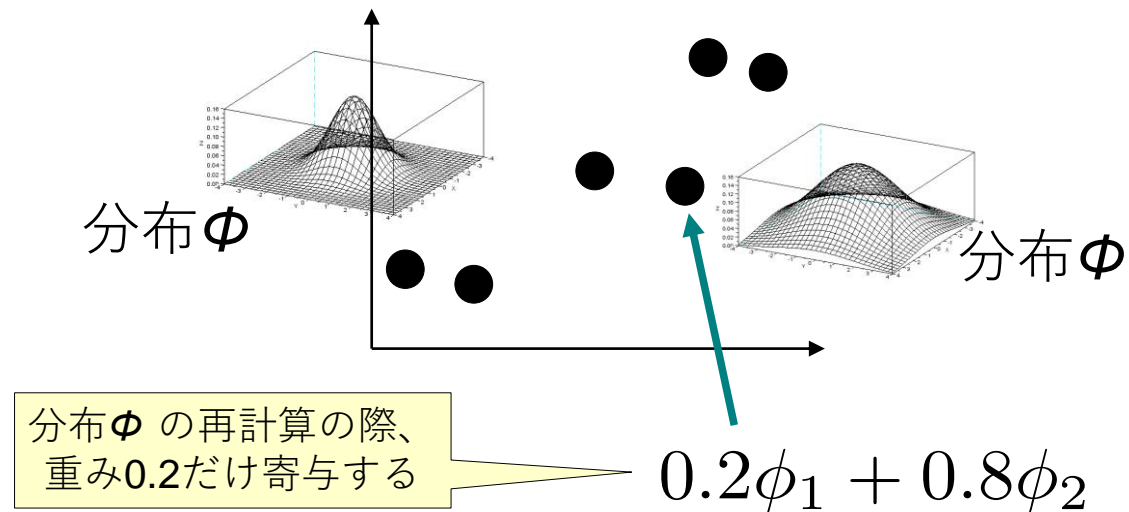
# 11.2.2 分割最適化クラスタリング

- k-means法の問題点 2
  - ◆ 得られる結果が初期値に大きく依存する
- 解決法 ⇒ Affinity Propagation アルゴリズム
  - ◆ すべてのデータがクラスタ中心の候補
  - ◆ クラスタ中心らしさ (responsibility) とクラスタへの属しやすさ (availability) をデータ間で伝達して収束させる
  - ◆ クラスタ数を予め決める必要がない



## 11.4 確率密度推定

- 教師なし学習で識別器を作る問題
  - ◆ クラスタリング結果からは、1クラス1プロトタイプのみ  
純な識別器しかできない
  - ◆ 各クラスの事前確率や確率密度関数も推定したい  
→ ガウス混合分布モデル



## 11.4 確率密度推定

- k-means法からガウス混合分布モデルへ（EMアルゴリズム）
  - ◆  $k$ 個の平均ベクトルを乱数で決める
    - ⇒  $k$ 個の正規分布を乱数で決める
  - ◆ 平均ベクトルとの距離を基準に、各データをいずれかのクラスタに所属させる
    - ⇒ 各分布が各データを生成する確率を計算し、各クラスタにゆるやかに所属させる
  - ◆ 所属させたデータをもとに平均ベクトルを再計算
    - ⇒ 各データのクラスタへの所属度に基づき各分布のパラメータ（平均、共分散行列）を再計算

## 11.4 確率密度推定

---

### Algorithm 11.3 EM アルゴリズム

---

入力: 正解なしデータ  $D$

出力: 各クラスを表す確率密度関数のパラメータ

入力空間上に  $k$  個のクラスタ  $c_j$  の分布  $\phi_j$  をランダムに設定

**repeat**

$/*$  E ステップ  $*/$

**for all**  $x_i \in D$  **do**

$\phi_j$  を用いて確率  $p(c_j | x_i)$  ( $j = 1, \dots, k$ ) を計算

**end for**

$/*$  M ステップ  $*/$

    E ステップで求めた  $p(c_j | x_i)$  を使って分布  $\phi_j$  のパラメータを再計算

**until** 分布のパラメータの変化量が閾値以下

**return**  $\phi_j$  ( $j = 1, \dots, k$ )

---



## 11.4 確率密度推定

- Eステップ：確率計算

$$\begin{aligned} p(c_m | \mathbf{x}_i) &= \frac{p(c_m)p(\mathbf{x}_i | c_m)}{p(\mathbf{x}_i)} \\ &= \frac{p(c_m)p(\mathbf{x}_i | c_m)}{\sum_{j=1}^k p(c_j)p(\mathbf{x}_i | c_j)} \\ &= \frac{p(c_m)\phi(\mathbf{x}_i; \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m)}{\sum_{j=1}^k p(c_j)\phi(\mathbf{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \end{aligned}$$

ベイズの定理

分母を周辺化

尤度を分布の式に置き換え

- Mステップ：分布の最尤推定

$$\begin{aligned} \boldsymbol{\mu}_m &= \frac{1}{|D|} \sum_{\mathbf{x}_i \in D} p(c_m | \mathbf{x}_i) \mathbf{x}_i \\ \boldsymbol{\Sigma}_m &= \frac{1}{|D|} \sum_{\mathbf{x}_i \in D} p(c_m | \mathbf{x}_i) (\mathbf{x}_i - \boldsymbol{\mu}_m)(\mathbf{x}_i - \boldsymbol{\mu}_m)^T \end{aligned}$$

## 11.4 確率密度推定

- ガウス混合分布モデルの問題点
  - ◆ 分割数 $k$ を予め決めなければならない
- 解決法  $\Rightarrow$  情報量規準の最小化
  - ◆ 2分割から始めて、分割数を適応的に決定する
  - ◆ 分割の妥当性の判断：BIC (Bayesian Information Criterion)が小さくなれば、分割を継続

$$BIC = -2 \log L + q \log N$$

少ないパラメータで尤度が高いものが良いモデル

- $L$ : モデルの尤度
- $q$ : モデルのパラメータ数
- $N$ : データ数

# まとめ

- モデル推定
  - ◆ データのまとまりを発見するプロセス
- 階層的クラスタリング
  - ◆ 類似度に基づいてボトムアップにデータをまとめてゆく
- 分割最適化クラスタリング
  - ◆ トップダウンでのデータの分割を最適化
- 確率密度推定
  - ◆ 分割最適化クラスタリングの一般化

補足

# Affinity Propagation

- データ $i$ とデータ $k$ の間に定義される3つの関数
  - ◆  $s(i,k)$  : データ $i$ とデータ $k$ の類似度。距離の反数がよく用いられる
  - ◆  $r(i,k)$  : 代表点 $k$ からデータ $i$ に送られるクラスタの代表点らしさ
  - ◆  $a(i,k)$  : データ $i$ から代表点 $k$ に送られるクラスタへの所属度

$r$ と $a$ が更新対象

# Affinity Propagationのアルゴリズム

1.  $a$ の値を0で初期化

2.  $r$ を以下の式で更新 ( $k'$ は競合している代表点候補)

$$r(i, k) \leftarrow s(i, k) - \max_{k' \text{ s.t. } k' \neq k} \{a(i, k') + s(i, k')\}$$

3.  $a$ を以下の式で更新 ( $i'$ は代表点候補  $k$ を最も高く評価している点)

$$a(i, k) \leftarrow \min\{0, r(k, k) + \sum_{i' \text{ s.t. } i' \notin \{i, k\}} r(i', k)\} \quad a(k, k) \leftarrow \sum_{i' \text{ s.t. } i' \neq k} \max\{0, r(i', k)\}$$

4.  $r_t$ と $a_t$ を用いて $r_{t+1}$ と $a_{t+1}$ を以下の式で更新

$$r_{t+1}(i, k) = \lambda r_t(i, k) + (1 - \lambda) r_{t+1}(i, k)$$

$$a_{t+1}(i, k) = \lambda a_t(i, k) + (1 - \lambda) a_{t+1}(i, k)$$

更新量があまり大きく  
ならないように

5. 更新量が一定値以下になれば終了。各データ $i$ を  $\operatorname{argmax}_k r(i, k) + a(i, k)$  となるデータ $k$ を代表とするクラスに割り当てる

# 11.3 異常検出

- 異常検出とは
  - ◆ 正常クラスの日ータと、それ以外のデータとのクラスタリング
  - ◆ 外れ値検知：学習データ中の異常値を検出
  - ◆ 新規検知：新しく入力されるデータの異常性を発見
- 外れ値検知（静的異常検出）
  - ◆ データの分布から大きく離れている値を見つける
  - ◆ 手法
    - 近くにデータがないか、あるいは極端に少ないものを外れ値とみなす
    - 「近く」の閾値を、予め決めておくことは難しい

# 11.3 異常検出

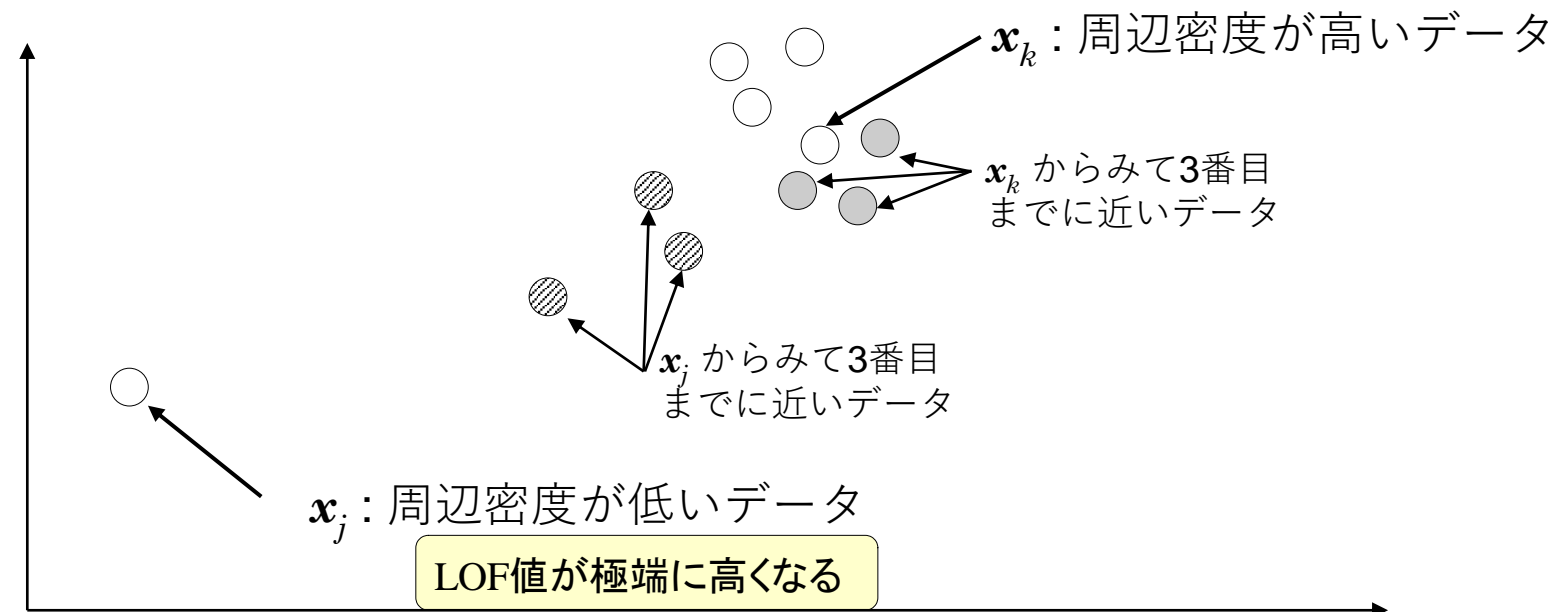
- 局所異常因子による外れ値検知

- ◆ 周辺密度

- あるデータの周辺の他のデータの集まり具合

- ◆ 局所異常因子(LOF: local outlier factor)

- あるデータの周辺密度と、その近くのk個のデータの周辺密度の平均との比





## 11.3 異常検出

- 局所異常因子の計算

- ◆ 到達可能距離

$$RD_k(\mathbf{x}, \mathbf{x}') = \max(\|\mathbf{x} - \mathbf{x}^{(k)}\|, \|\mathbf{x} - \mathbf{x}'\|)$$

$\mathbf{x}^{(k)}$  は、 $\mathbf{x}$  に  $k$  番目に近いデータ

近すぎる距離は、 $k$  番目との距離に補正される

- ◆ 局所到達可能密度

$$LRD_k(\mathbf{x}) = \left( \frac{1}{k} \sum_{i=1}^k RD_k(\mathbf{x}^{(i)}, \mathbf{x}) \right)^{-1}$$

$\mathbf{x}$  の周りの密度が高い場合、大きな値になる

- ◆ 局所異常因子

$$LOF_k(\mathbf{x}) = \frac{\frac{1}{k} \sum_{i=1}^k LRD_k(\mathbf{x}^{(i)})}{LRD_k(\mathbf{x})}$$

# One-class SVM

- One-class SVMによる新規検知
  - ◆ RBFカーネルによる写像後の空間における学習データを正例、原点を負例とみなして境界を得る
  - ◆ 新規データに対して、境界の外の場合は異常とみなす

