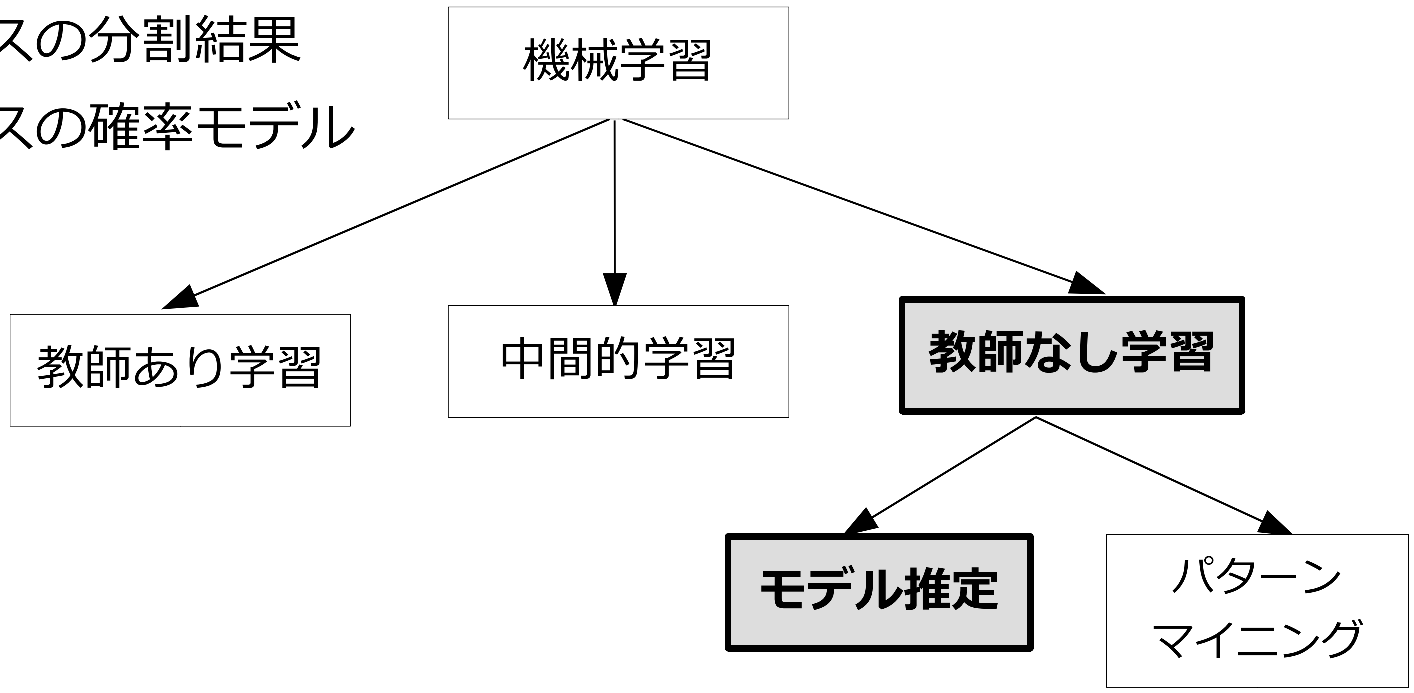
11. モデル推定

- 問題設定
  - 教師なし学習
  - 数値入力 → クラスモデル
    - クラスモデルの例
      - クラスの分割結果
      - クラスの確率モデル

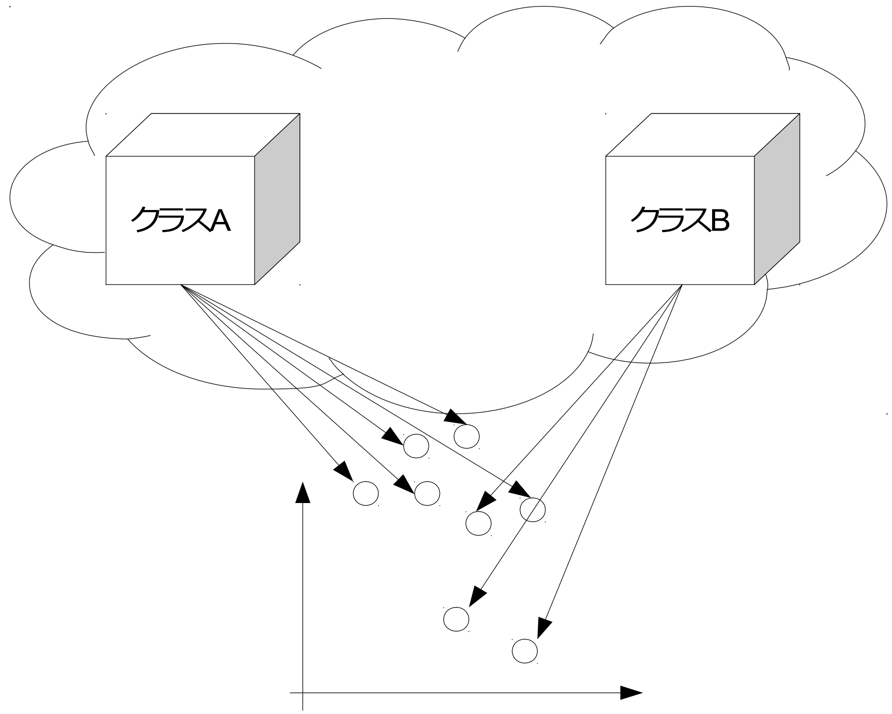


11.1 数値特徴に対する「教師なし・モデル推定」問題の定義

- 学習データ

$$\{\boldsymbol{x}^{(i)}\} \qquad i = 1, .., N$$

- 問題設定
  - 特徴ベクトル $\boldsymbol{x}$ が生成された元のクラスの性質を推定する



ID: 1103

11.2 クラスタリング

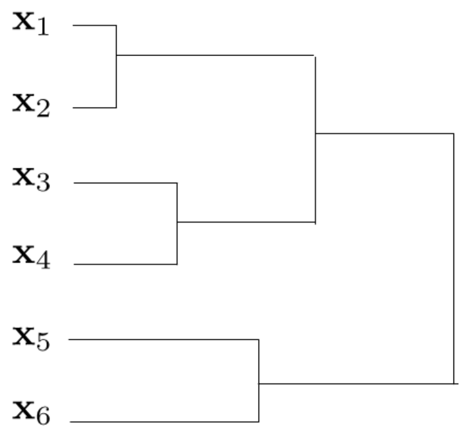
- クラスタリングとは
  - 対象のデータを、
    内的結合（同じ集合内のデータ間の距離は小さく）
    　　と
    外的分離（異なる集合間の距離は大きく）
    が達成されるような部分集合に分割すること

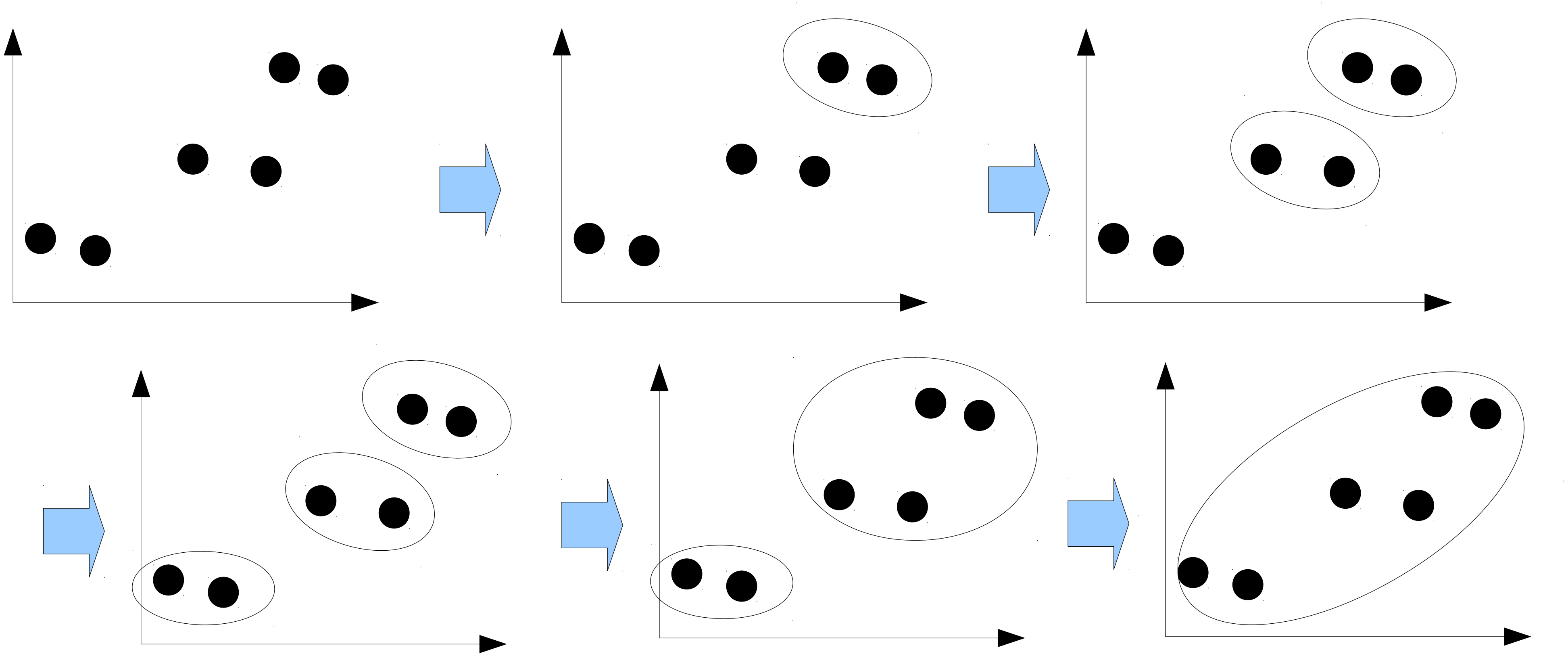
要するに
塊を見つ
けること

- 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. 1 データ 1 クラスタからスタート
  2. 最も近接するクラスタをまとめる
  3. 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 */

$$(c_m, c_n) \leftarrow \underset{c_i, c_j \in C}{\arg\max} \, \mathrm{sim}(c_i, c_j)$$

$\{c_m, c_n\}$ 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
  - 単連結法
    - 最も近い事例対の距離を類似度とする。
    - クラスタが一方向に伸びやすくなる傾向がある。
  - 完全連結法
    - 最も遠い事例対の距離を類似度とする。
    - クラスタが一方向に伸びるのを避ける傾向がある。
  - 重心法
    - クラスタの重心間の距離を類似度とする。
    - クラスタの伸び方は、単連結と完全連結の間
  - Ward 法
    - 融合前後の「クラスタ中心との距離の二乗和」の差

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、全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
    - 2 分割で 2^N 通り
  - 探索によって、準最適解を求める

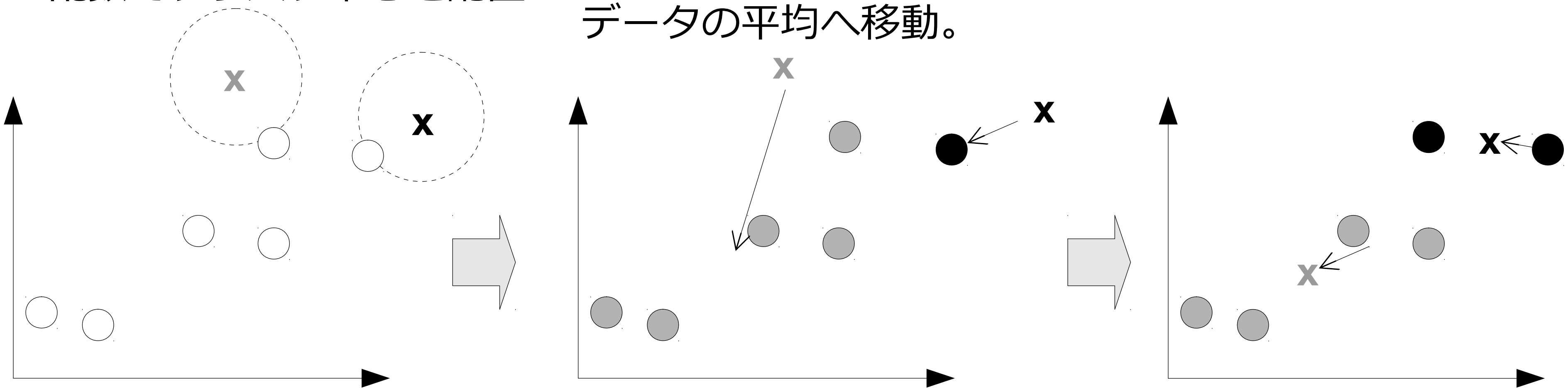
k-means アルゴリズム

- k-Means アルゴリズム
  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

k=2 とし、初期値として
乱数でクラスタ中心を配置

全データを近い方のクラスタ
中心に所属させる。そして、
クラスタ中心を所属している
データの平均へ移動。

左の処理を繰り返す。



k-means アルゴリズム

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 $\boldsymbol{\mu}_j \quad (j = 1, \ldots, k)$

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 $\boldsymbol{\mu}_j$ とする

repeat

  for all $\boldsymbol{x}_i \in D$ **do**

    各クラスタ中心 $\boldsymbol{\mu}_j$ 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) */

$$\boldsymbol{\mu}_j \leftarrow \frac{1}{N_j} \sum_{\boldsymbol{x}_k \in \text{クラスタ}_j} \boldsymbol{x}_k$$

until クラスタ中心 $\boldsymbol{\mu}_j$ が変化しない

return $\boldsymbol{\mu}_j \quad (j = 1, \ldots, k)$

自動で分割数を決定するクラスタリング

- k-means 法の問題点
  - 分割数 k を予め決めなければならない
- 解決法 ⇒ X-means アルゴリズム
  - 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

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

– L: モデルの尤度
– q: モデルのパラメータ数
– N: データ数

パラメータで表される
統計モデルの選択基準
(小さいほどよいモデル)

ID: 1111

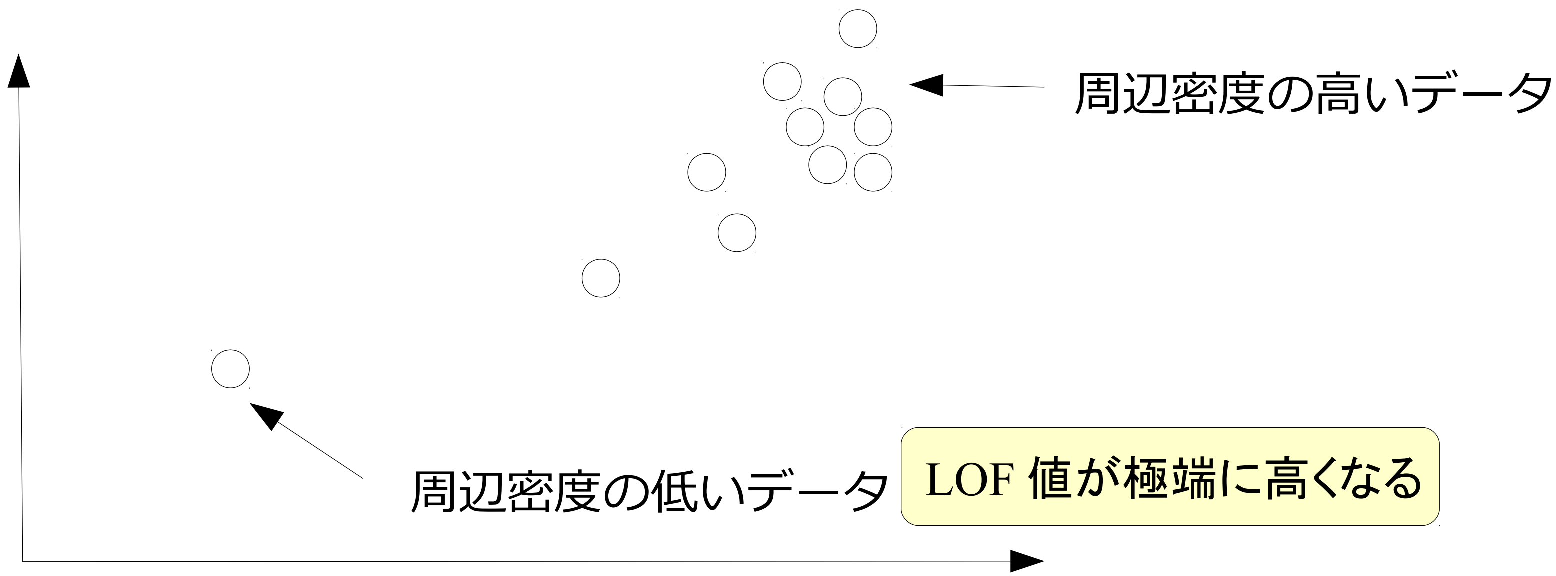
11.3 異常検出

- 異常検出とは
  - 正常クラスのデータと、それ以外のデータとのクラスタリング
  - 外れ値検知、変化点検出、異常状態検出など
  - 対象データが静的・動的で手法が異なる
- 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

ID: 1112

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)
    - 近くの k 個のデータの周辺密度の平均と、あるデータの周辺密度との比



11.3 異常検出

- 局所異常因子の計算

  - 到達可能距離

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

$\boldsymbol{x}^{(k)}$ は、$\boldsymbol{x}$ 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

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

x の周りの密度が高い場合、大きな値になる

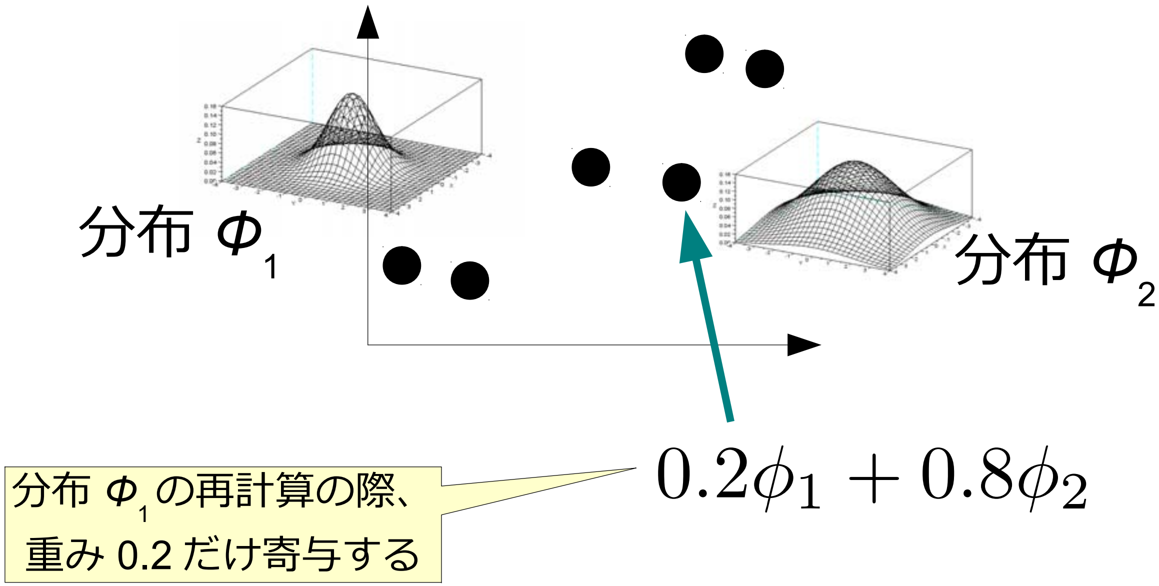
  - 局所異常因子

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

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プロトタイプの単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

⇨ EM アルゴリズム



11.4 確率密度推定

- k-means 法の一般化
  - 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帰属させる
  - 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帰属度に基づき各分布のパラメータ（平均値、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 個の分布 ϕ_j をランダムに設定

repeat

/* E ステップ */

for all 学習データ $\boldsymbol{x}^{(i)}$ **do**

$p(\boldsymbol{x}^{(i)}|c_j) = \phi_j(\boldsymbol{x}^{(i)}) \quad (j = 1, \ldots, k)$ を計算

end for

/* M ステップ*/

E ステップの確率 $p(\boldsymbol{x}^{(i)}|c_j)$ を使って分布 ϕ_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