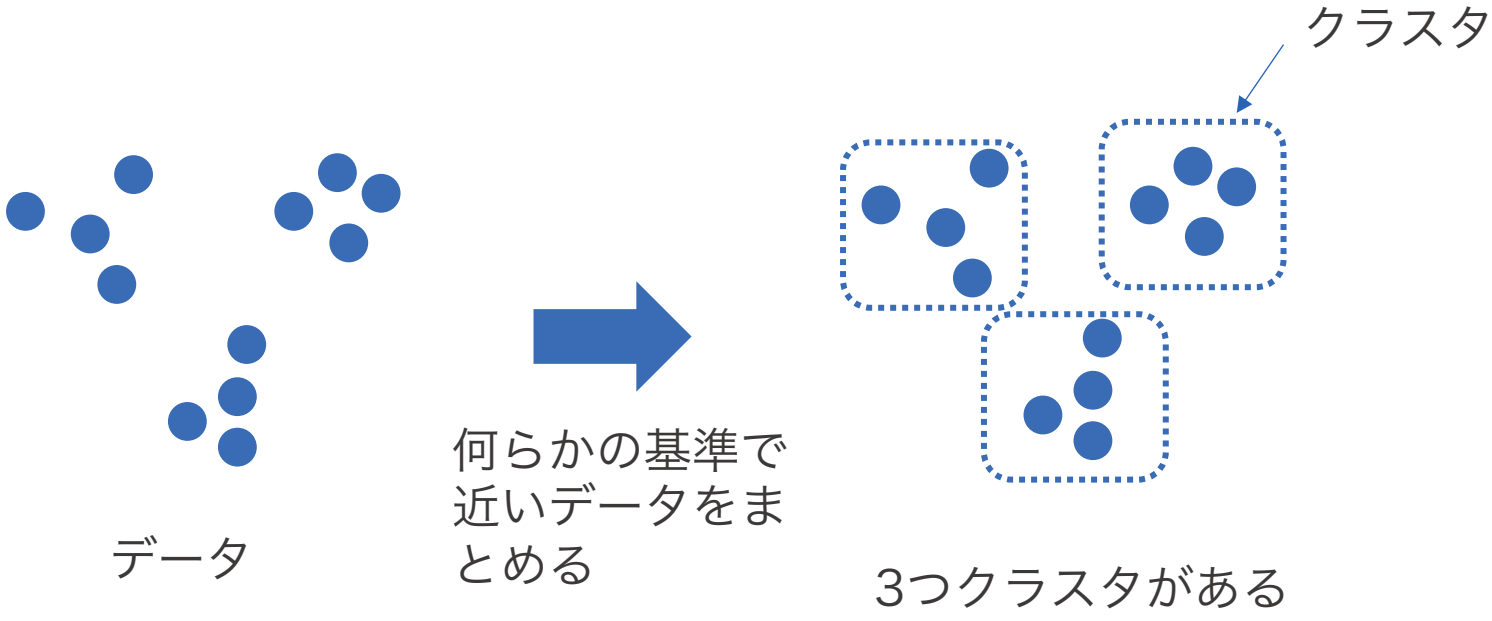
クラスタリング

Spectral clusterinとクラスタ数推定は作成中

公立小松大学
藤田 一寿

クラスタリングとは

- データをクラスタに自動で分けること
  - クラスタとはデータを近い遠いでグループ分けしてできたグループのこと.
  - 近い遠いを決める基準は様々にある.
  - クラスタの意味は後で考える.

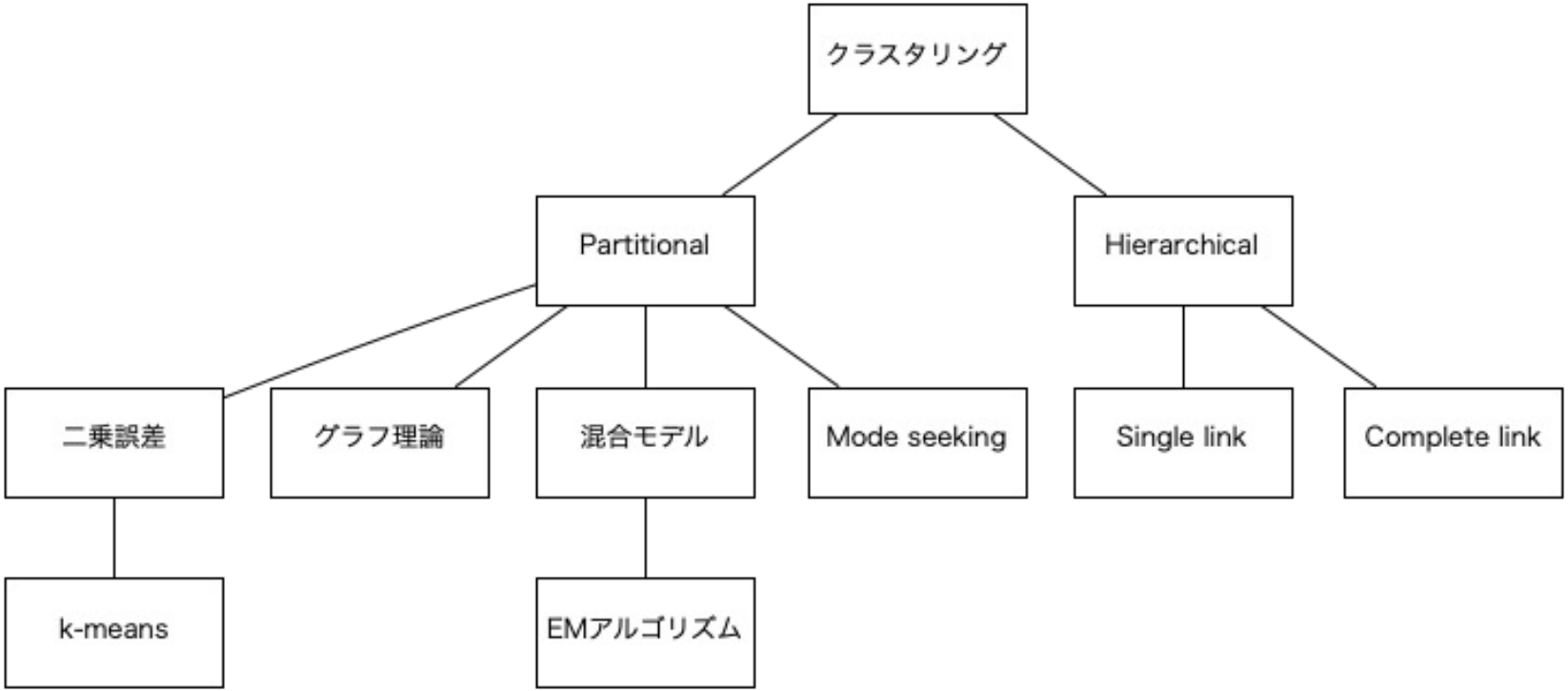


例：クラスタリングによる領域分割

- 画像を色の類似性により領域を分割する
- 類似した色をまとめるときにクラスタリングを用いる

似た色をまとめる

クラスタリング手法の分類



似てる似てないの基準

- 機械は入力が似ている似ていないを判別するには基準が必要

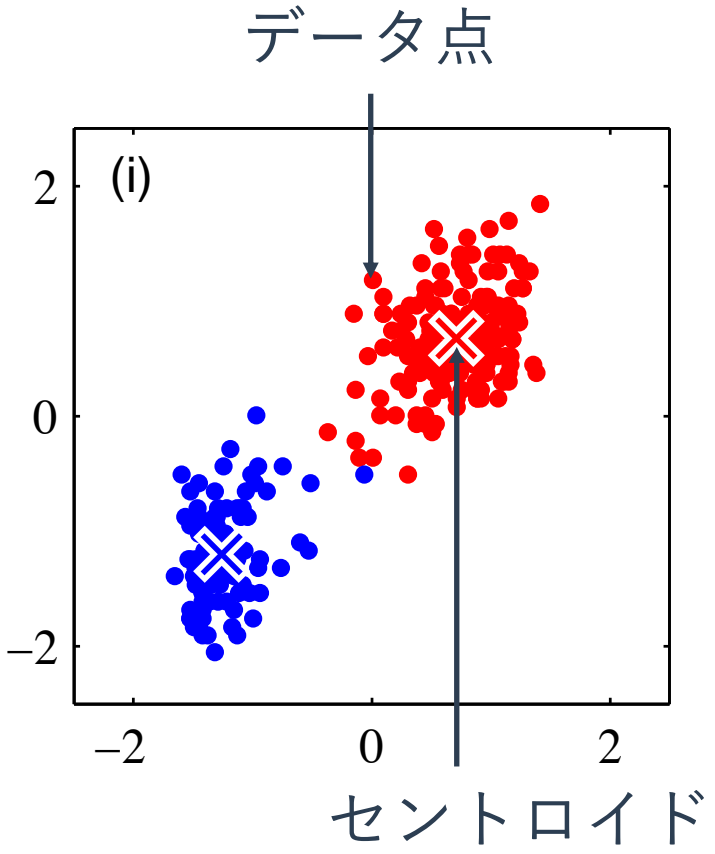
- 判断基準の例
  - ユークリッド距離
    - $d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots} = \|\boldsymbol{x} - \boldsymbol{y}\| = \|\boldsymbol{x} - \boldsymbol{y}\|_2$
  - ミンコフスキー距離
    - $d(\boldsymbol{x}, \boldsymbol{y}) = (|x_1 - y_1|^p + |x_2 - y_2|^p + \cdots)^{1/p} = \|\boldsymbol{x} - \boldsymbol{y}\|_p$
  - マハラノビス距離
    - $d(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^{\mathrm{T}} \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{y})}$
    - Σは共分散行列
  - コサイン類似度
    - $S(\boldsymbol{x}, \boldsymbol{y}) = \cos\theta = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\|\boldsymbol{x}\| \|\boldsymbol{y}\|}$

距離：似ていれば似ているほど小さい値
類似度：似ていれば似ているほど大きな値

k-means

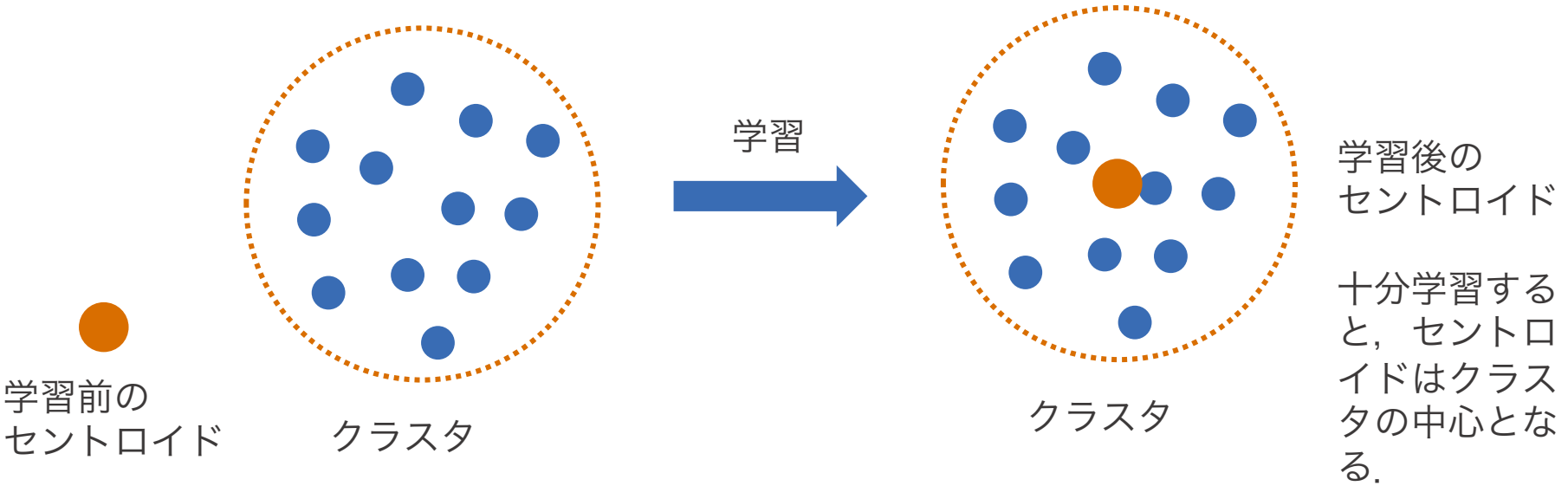
用語の復習

- データ点
  - データセットに含まれる1つのデータ
- クラスタ
  - データ点の集まり
- セントロイド
  - クラスタの中心



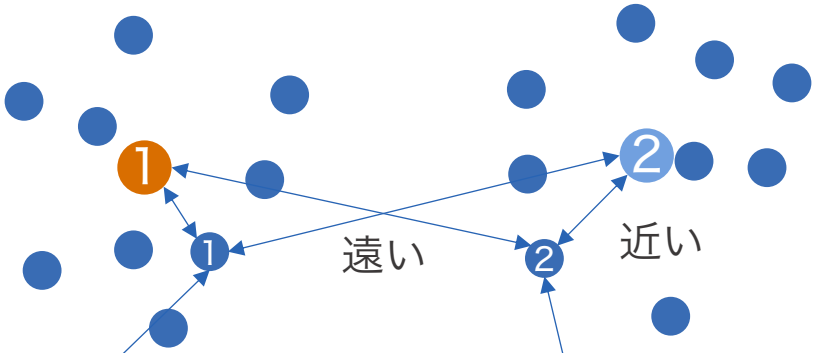
k-means

- ユークリッド距離を基準にクラスタリングする手法
- k-meansでは，クラスタの中心である「セントロイド」と呼ばれるベクトルを求めながらクラスタリングを行う.



クラスタの決め方

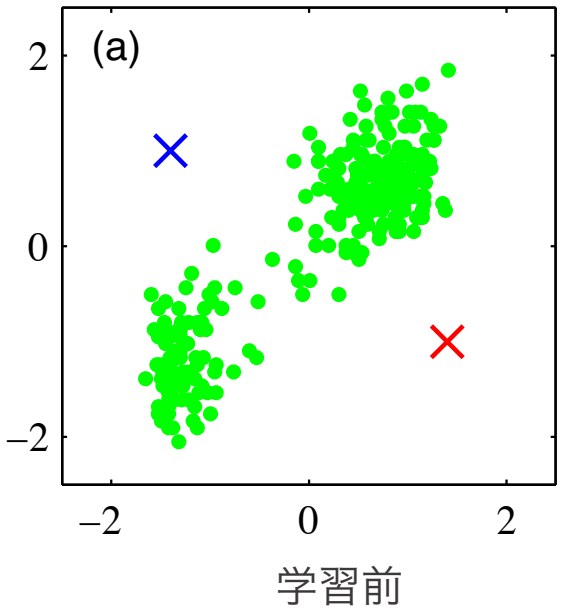
- 学習したセントロイドからクラスタを決める.

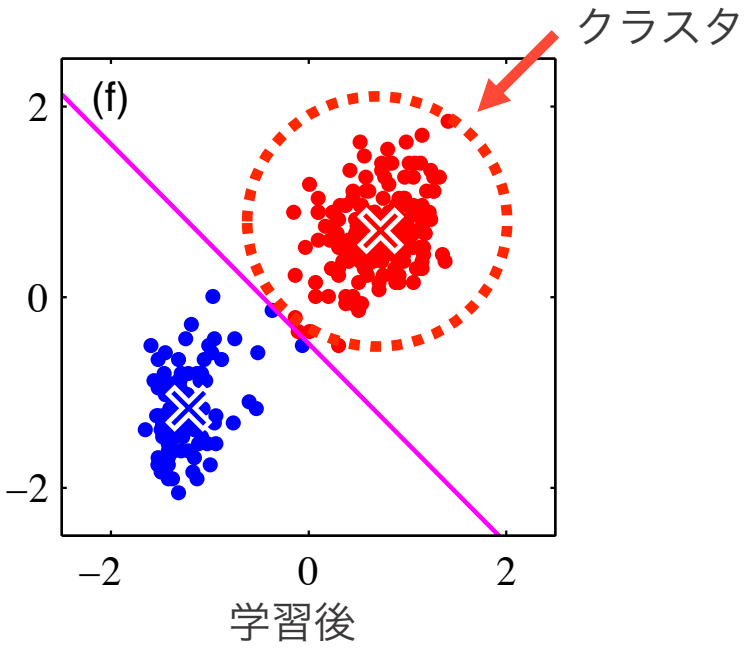


このデータ点はセントロイド2よりセントロイド1の方に近いためクラスタ1に所属する.

このデータ点はセントロイド2よりセントロイド1の方に近いためクラスタ1に所属する.

クラスタリング結果





Xがセントロイドを表している.
学習前はセントロイドはランダムに初期化されるため，クラスタの中心にない.
学習後はセントロイドはクラスタの中心に移動し，データ点もクラスタに所属している.

k-meansの目的関数

- 良いセントロイドと良いクラスタリング結果はどういうものか
- データ点とそのデータ点が所属するクラスタのセントロイドとの距離の総和が最小になるとき最適なクラスタリング結果が得られると考える.
- データ点とそのデータ点が所属するクラスタのセントロイドとの距離の総和は,

データ点

$$J = \sum_{i}^{N} \sum_{j}^{K} r_{ij} \|\boldsymbol{x}_i - \boldsymbol{m}_j\|^2$$

セントロイド

- $\boldsymbol{x}_i$はデータ点のベクトル, $\boldsymbol{m}_j$はクラスタjのセントロイド, r_{ij}はデータ点iがクラスタjに所属していれば1, そうでなければ0となる変数.
- この式を最小にするよう学習すれば良い. 学習を通し最大化もしくは最小化したい関数のことを目的関数という.

k-meansのアルゴリズム

- k-meansの場合，先に示した目的関数を最小化することでクラスタリングを達成する．最小化するアルゴリズムは次のとおりである．

- k-meansのアルゴリズム

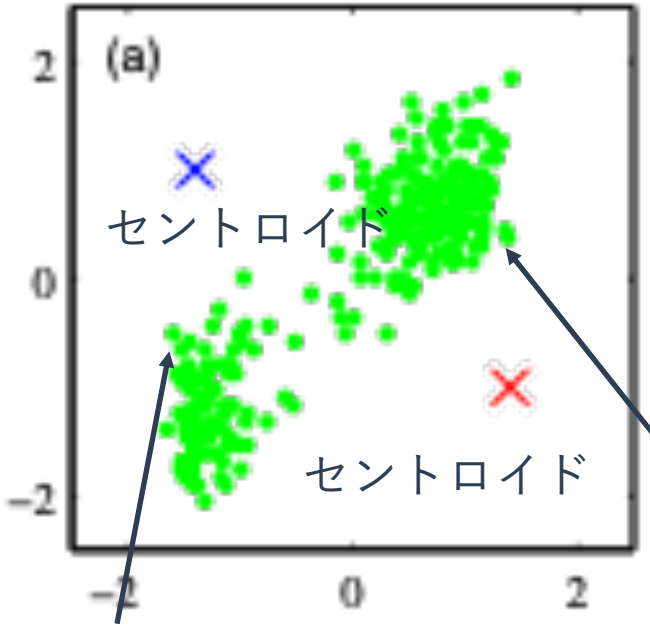
1. クラスタ数k，セントロイドm_1，…，m_j，…，m_kを初期化
2. すべてのデータ点を最も近いセントロイドを持つクラスタに所属させる
3. セントロイドc_jを次の式で計算する

$$\boldsymbol{c}_j = \frac{1}{\sum_i r_{ij}} \sum_i r_{ij} \boldsymbol{x}_i$$

4. 各クラスタに所属するデータ点の個数が変化しなかった場合もしくは目的関数の値が収束した場合もしくは指定の回数2，3を実行した場合は終了，そうでない場合2に戻る

処理2：データ点をクラスタに所属させる

初期状態

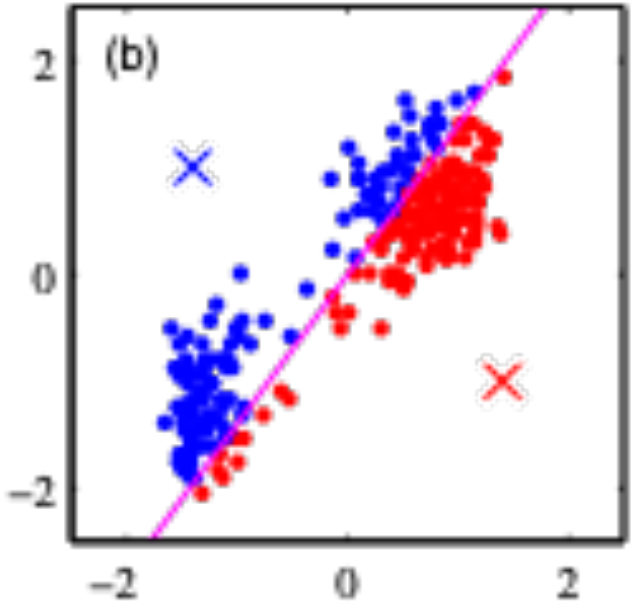


セントロイド

セントロイド

処理2

すべてのデータ点について
クラスタを決定した結果

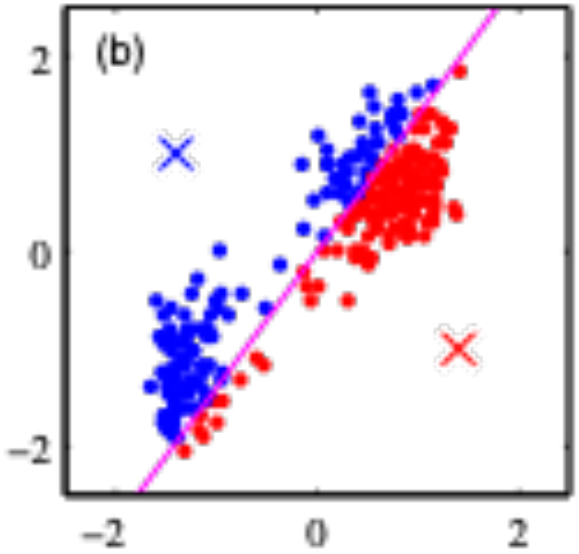


このデータ点は青xに近い
ので青のクラスタに所属

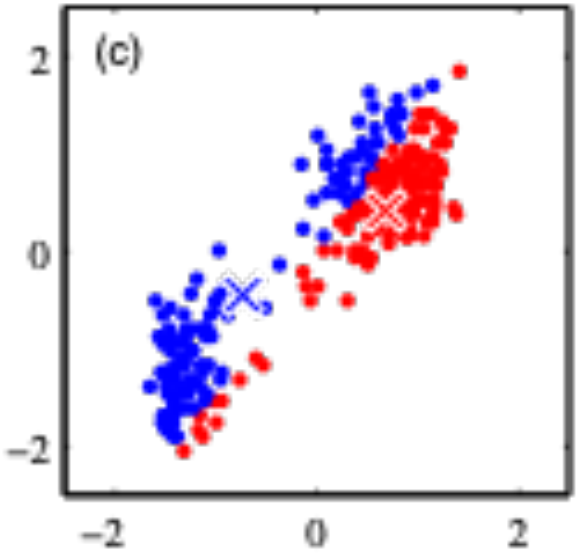
このデータ点は赤xに近いの
で赤のクラスタに所属

処理３：セントロイドの更新

処理2を適用した結果

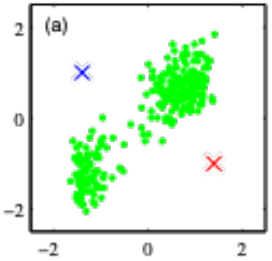


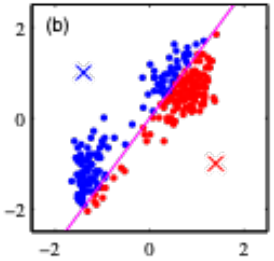
セントロイドを計算した結果



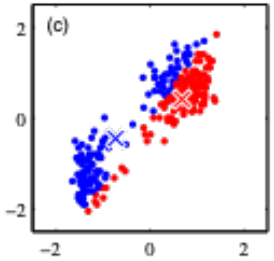
処理2で得られたクラスタに所属するデータ点の平均がセントロイドとなる・

k-meansの学習過程

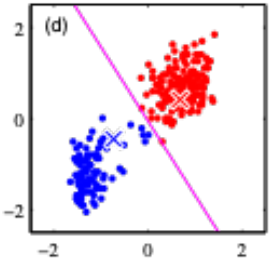




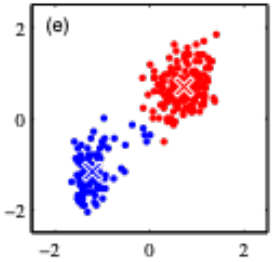
処理2



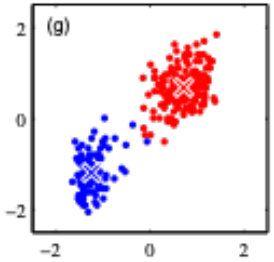
処理3



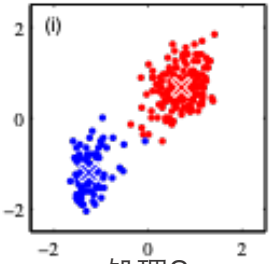
処理2



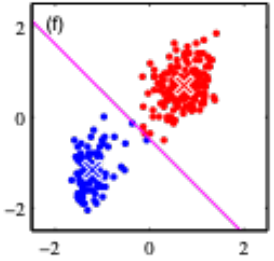
処理3



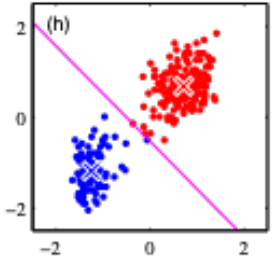
処理2



処理2



処理3



処理2

k-meansによる領域分割（減色）の例

k-meansの利点と欠点

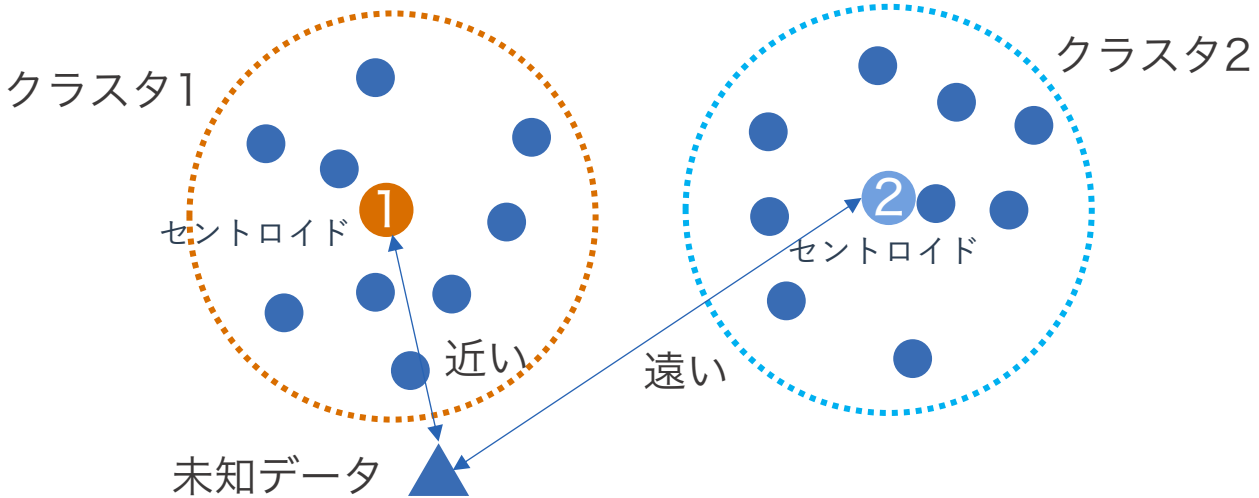
- 利点
  - 実装が簡単である.
  - アルゴリズムが分かりやすい.
  - 結果を理解しやすい.
    - k-meansは空間をVoronoi cellに分ける.
- 欠点
  - 外れ値に弱い.
  - クラスタリング結果が初期値に依存する.
    - k-means++で改善
  - データの分布が等方ガウス分布のときにのみ適切にクラスタリングできる（何が適切なのかは難しい問題だが）

k-meansの改変

- k-meansでは，近いデータを同じクラスタとしてグループ分けした.
- k-meansでは遠い近いの基準にユークリッド距離を用いた.
  - $d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sqrt{(\boldsymbol{x}_1 - \boldsymbol{x}_2)^{\mathrm{T}}(\boldsymbol{x}_1 - \boldsymbol{x}_2)} = \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$
- 基準として別のものを用いることも可能である.
  - コサイン類似度
    - $S(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{\boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{x}_2}{\|\boldsymbol{x}_1\|\|\boldsymbol{x}_2\|}$
    - コサイン類似度を使用した場合目的関数を最大化する.
  - マハラノビス距離
    - $d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \left((\boldsymbol{x}_1 - \boldsymbol{x}_2)^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{x}_1 - \boldsymbol{x}_2)\right)^{1/2}$
  - ミンコフスキー距離
    - $d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \left(\sum_i^D (x_{1i} - x_{2i})^p\right)^{1/p}$

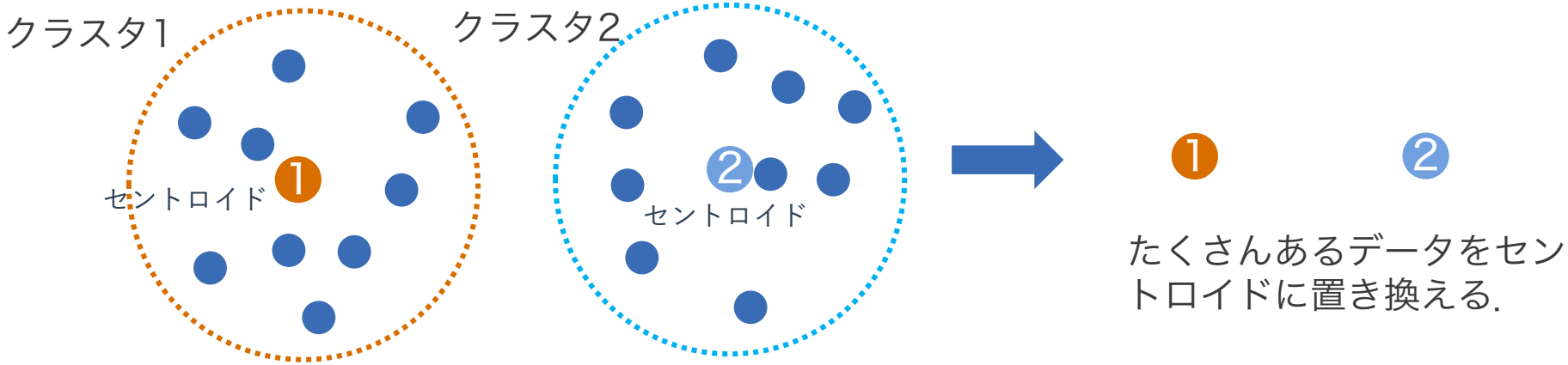
k-meansの応用

- クラスタリング
  - k個のクラスタにデータを分けることが可能
  - データの類似性を調べるために使用できる



k-meansの応用

- 量子化
  - k-meansによりk個のセントロイドを得ることができる.
  - k-meansはデータをk個のセントロイドベクトルに代表させたと言える.
  - 見方を変えれば，データをk個のベクトルに圧縮したといえる.
  - データを少数のベクトルに置き換えることを量子化（ベクトル量子化）という.



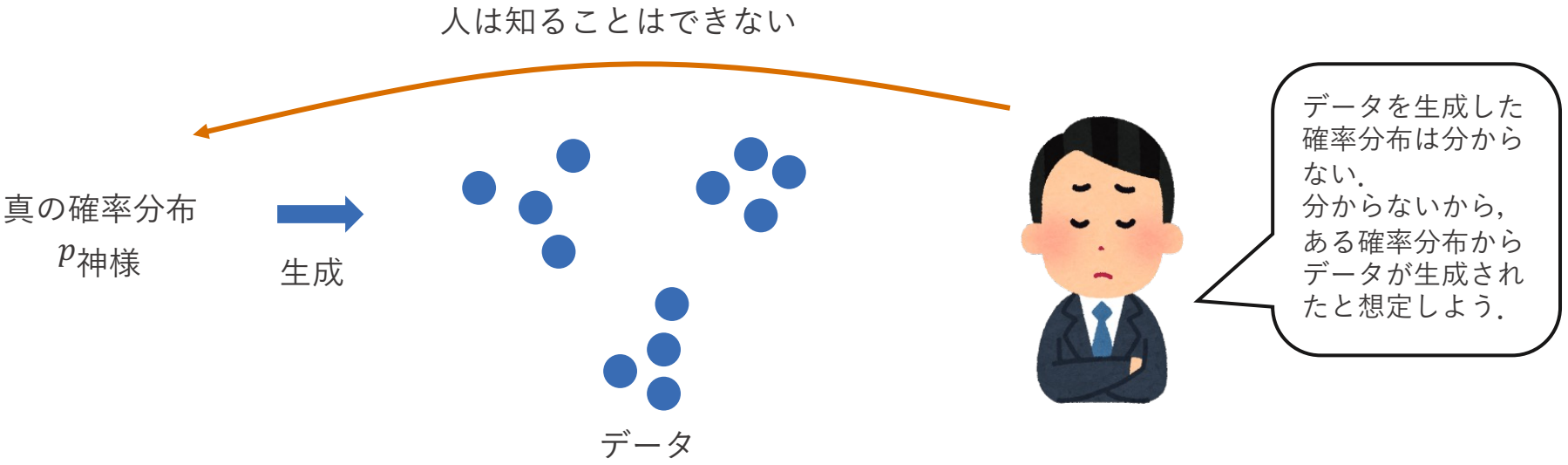
混合ガウス分布 (Gaussian mixture model)

model based clustering

- データが何かの混合分布から生成されたと考えてクラスタリングする方法をmodel based clusteringという.
- 混合ガウス分布を想定することがほとんど(Gaussian Mixture Model: GMM)
- model based clusteringでは混合分布のパラメタの推定問題となる.

分布とモデル

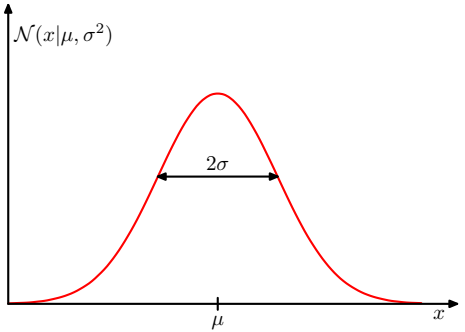
- データはある確率分布から生じる．その確率分布は神様しか知らない．
- 我々はそのデータがどのような確率分布から生成されたか想定（想像）することはできる．
- 想定した確率分布をモデルという．



ガウス分布

- 釣鐘型の分布をガウス分布（正規分布）という.
- 最もよく用いられる確率分布である.

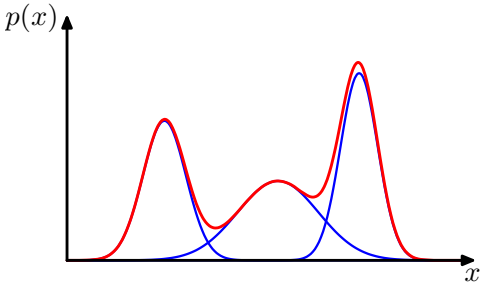
- 1次元ガウス分布
  - $N(x \mid \mu, \sigma) = \frac{1}{(2\pi\sigma)^{1/2}} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right)$
  - μは平均，　σは分散である.



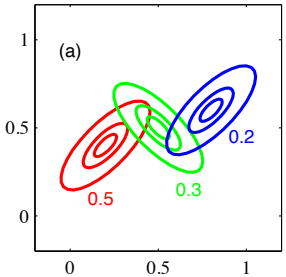
- D次元ガウス分布
  - $N(\boldsymbol{x} \mid \boldsymbol{\mu}, \Sigma) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\mathrm{T}}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right)$
  - $\boldsymbol{\mu}$はD次元の平均ベクトル，　Σは$D \times D$の共分散行列，$|\Sigma|$はΣの行列式である.

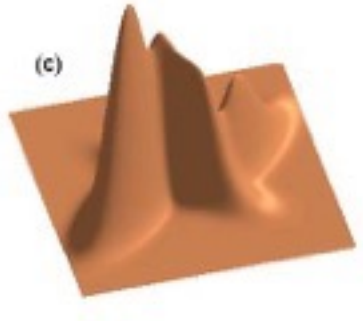
混合ガウス分布(GMM: Gaussian Mixture Model)

- 複数のガウス分布からなる確率分布を混合ガウス分布という.
- $p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k N(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \Sigma_k)$
- Kはガウス分布の数，Nはガウス分布，π_kは混合係数，Σ_kは共分散行列を表す.
- データが複数のガウス分布で出ていると仮定したときの確率モデルを混合ガウスモデルという.



1次元混合ガウス分布の例
3つのガウス分布が混合されている.





2次元混合ガウス分布の例
3つのガウス分布が混合されている.
左図は個々のガウス分布の様子とそれぞれの混合係数を表す.
右図は混合ガウス分布の確率分布を表す.

混合係数

- 混合係数はすべて足すと1になる.
- $\sum_{k=1}^{K} \pi_k = 1$
- つまり，π_kは確率とみなせる.

確率変数z

- ここで，$\boldsymbol{z} = \{z_1, \ldots, z_k, \ldots, z_K\}$，$z_k = \{0,1\}$，$\sum_k z_k = 1$という確率変数を導入する．
- $\sum_k z_k = 1$だから，$z_k = 1$ならば，それ以外の要素は 0 である（1-of-k coding）．
- $p(z_k = 1) = \pi_k$
- とする．これを書き方を変えると
- $p(\boldsymbol{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$
- と書ける．
- zの値が与えられた下での$\boldsymbol{x}$の確率分布は
- $p(\boldsymbol{x} \mid z_k = 1) = N(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
- これも書き直すと
- $p(\boldsymbol{x} \mid \boldsymbol{z}) = \prod_{k=1}^{K} N(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}$
- と書ける．
- 同時確率$p(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{z}) = p(\boldsymbol{x}, \boldsymbol{z})$を$\boldsymbol{z}$について周辺化したものが混合ガウス分布となる．
- $p(\boldsymbol{x}) = \sum_z p(\boldsymbol{x} \mid \boldsymbol{z})p(\boldsymbol{z}) = \sum_{k=1}^{K} \pi_k N(\boldsymbol{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$

$\prod_k \pi_k^{z_k} = 1 \times \cdots \times \pi_k \times \cdots \times 1$となる．
何故ならば$z_k = 0$ならば$\pi_k^{z_k} = 1$だからである．

これはガウス分布kのみを想定したときの$\boldsymbol{x}$が出てくる確率とみなせる．

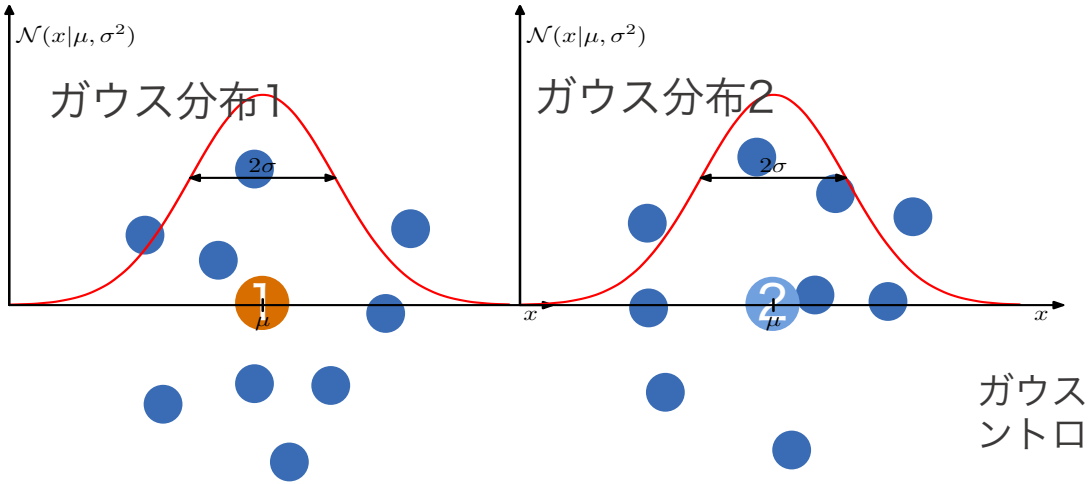
負担率

- 逆に，$\boldsymbol{x}$が観測されたとき，それがガウス分布kから生成された確率を考える．ここでベイズ定理を用いて
- $p(z_k = 1 \mid \boldsymbol{x}) = \frac{p(z_k=1)p(\boldsymbol{x}|z_k = 1)}{p(\boldsymbol{x})} = \frac{\pi_k N(\boldsymbol{x}|\boldsymbol{\mu}_k,\Sigma_k)}{\sum_{i=1}^{K} \pi_i N(\boldsymbol{x}|\boldsymbol{\mu}_i,\Sigma_i)}$
- と書ける．
- これを負担率という．
- 負担率は要素kが$\boldsymbol{x}$の観測を説明する度合いと解釈できる．

GMMを使ったクラスタリング

- データは混合ガウス分布から生成されたと仮定する.
- 混合ガウス分布を構成するガウス分布は，データがあるクラスタから生成される確率を表すとする.
- 逆にデータ点が生成される可能性が最も高いガウス分布で表されるクラスタにデータ点は所属すると考えられる.

クラスタ1はガウス分布1から生じたと考える.



クラスタ2はガウス分布2から生じたと考える.

ガウス分布の平均がセントロイドとなる.

最尤推定

- GMMを用いたクラスタリングは，GMMのパラメタを求めることである．
- GMMのパラメタを求めるのに最尤推定を用いる．
- データ集合$\mathrm{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$があり，これらが互いに独立に生成されたと仮定する (i.i.d.: independent and identically distributed)．
- このとき，log尤度は次のように書ける．
- $\ln p(X \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k N(\boldsymbol{x}_n \mid \mu_k, \Sigma_k)\}$
- これを最大にするパラメタを最適解とし，このパラメタを探す最尤推定という．
- しかし，解析的に求めることは困難である．

$p(X \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$はパラメタ$\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$のときデータ集合Xが出現する確率である．この確率が最も高いパラメタが良いパラメタということになる．この確率を尤度という．logは単調増加関数なので尤度の対数の最大値を取るパラメタは，尤度の最大値を取るパラメタと一致する．

最尤推定

- 関数が最大値のとき微分は0の極値となる．尤度関数が最大のパラメタのとき，尤度関数の微分は0となることが予想される．
- まず，平均$\boldsymbol{\mu}_k$について尤度関数を微分する．

$$\frac{\partial \ln p(X \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{\partial \boldsymbol{\mu}_k} = \frac{\partial}{\partial \boldsymbol{\mu}_k} \sum_{n=1}^{N} \ln \left\{ \sum_{i=1}^{K} \pi_i N(\boldsymbol{x}_n \mid \mu_i, \Sigma_i) \right\}$$

$$= \sum_{n=1}^{N} \frac{\pi_k N(\boldsymbol{x}_k \mid \mu_k, \Sigma_k)}{\sum_{i=1}^{K} \pi_i N(\boldsymbol{x}_n \mid \mu_i, \Sigma_i)} \Sigma_k^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0$$

logの微分$(\log f(x))' = \frac{f'(x)}{f(x)}$

合成関数の微分$\Big(f\big(g(x)\big)\Big)' = f'\big(g(x)\big)g'(x)$

- $\frac{\pi_k N(\boldsymbol{x}_k|\mu_k, \Sigma_k)}{\sum_{i=1}^{K} \pi_i N(\boldsymbol{x}_n|\mu_i, \Sigma_i)} = \gamma(z_{nk})$は負担率なので
- $\sum_{n=1}^{N} \gamma(z_{nk}) \Sigma_k^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0$
- と書ける．この式を満たす平均$\boldsymbol{\mu}_k$を求めれば良い．

最尤推定

- $\sum_{n=1}^{N} \gamma(z_{nk}) \Sigma_k^{-1} (\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0$
- を解く． Σ_k を両辺に書けると$\Sigma_k \, \Sigma_k^{-1} = I$となり$\Sigma_k^{-1}$は消えるとして整理すると，

逆行列がないと計算できない…

- $\mu_k = \dfrac{\sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n}{\sum_{n=1}^{N} \gamma(z_{nk})}$
- $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$とおくと

N_Kはクラスタkに割り当てられた実質的な数と解釈できる.

- $\mu_k = \dfrac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \, \boldsymbol{x}_n$
- 同様に微分が0となる共分散を求めると

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\boldsymbol{x}_n - \boldsymbol{\mu}_k) (\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

最尤推定

- 最後に，混合係数π_kを求める．混合係数には総和が1という制約条件がある．この制約条件のもと尤度関数を最大化しなければならない．
- このようなとき，ラグランジュの未定乗数法を用いる．
- $L = \ln p(X \mid \pi, \boldsymbol{\mu}, \Sigma) + \lambda\left(\sum_{k=1}^{K} \pi_k - 1\right)$
- という関数を考え，これを微分して0となるπ_kを求める．そうすると，
- $\sum_{n=1}^{N} \frac{N(\boldsymbol{x}_k|\mu_k, \Sigma_k)}{\sum_{i=1}^{K} \pi_i N(\boldsymbol{x}_n|\mu_i, \Sigma_i)} + \lambda = 0$
- 両辺π_kを書けると，sumは負担率となる．

最尤推定

- $\sum_{n=1}^{N} \frac{\pi_k N(\boldsymbol{x}_k|\mu_k, \Sigma_k)}{\sum_{i=1}^{K} \pi_i N(\boldsymbol{x}_n|\mu_i, \Sigma_i)} + \lambda\pi_k = \sum_{n=1}^{N} \gamma(z_{nk}) + \lambda\pi_k = 0$
- これを計算すると
- $\lambda = -N$
- を得る．これを用いると
- $\pi_k = \frac{N_k}{N}$
- となる．

$$\sum_{n=1}^{N} \gamma(z_{nk}) + \lambda\pi_k = 0$$

$$\lambda\pi_k = -N_k$$

$$\lambda \sum_{k=1}^{K} \pi_k = -\sum_{k=1}^{K} N_k$$

$$\lambda = -N$$

$$\sum_{n=1}^{N} \gamma(z_{nk}) - N\pi_k = 0$$

$$-N\pi_k = -N_k$$

$$\pi_k = \frac{N_k}{N}$$

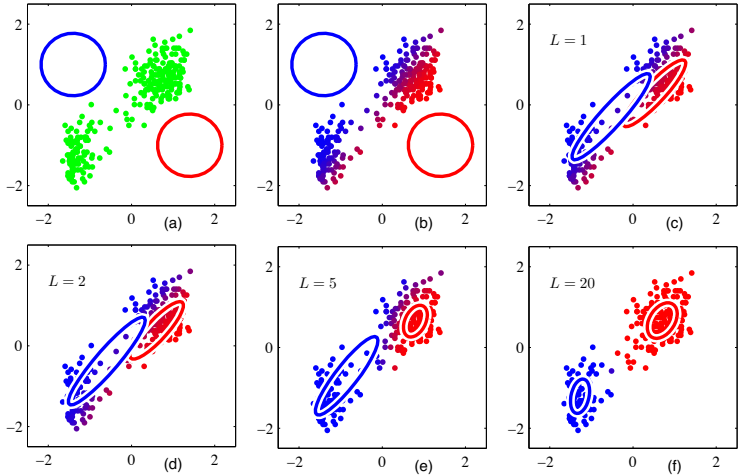
EM アルゴリズム

- GMMでクラスタリングを行うためには混合ガウス分布のパラメタを決める必要がある．しかし，解析的に求めることは困難である．
- ここでEMアルゴリズムを用い，尤度関数を最大にするパラメタを探すことにする．

- 混合ガウス分布を用いたクラスタリングでは，EMアルゴリズムを用い混合ガウス分布のパラメタを求める．
  - GMMを用いたクラスタリングは，EMアルゴリズムを用いGMMのパラメタを求めることである．

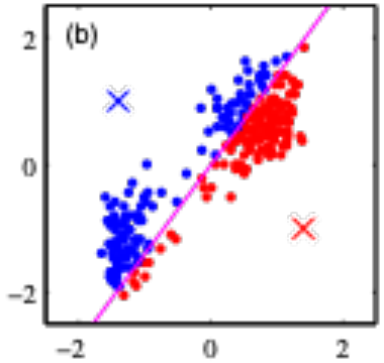
EMアルゴリズム

1. データ集合$X = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$があるとする.
2. パラメタ$\boldsymbol{\mu}_k, \Sigma_k, \pi_k$を適当な数値で初期化する.
3. 現在のパラメタで負担率を求める.
4. 3で計算した負担率を用い，パラメタ$\boldsymbol{\mu}_k, \Sigma_k, \pi_k$を計算する.
5. 終了条件を満たしていなければ3に戻る.

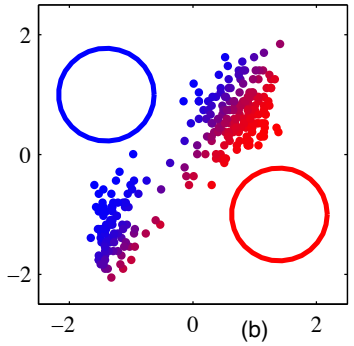


GMMとk-means

- k-meansは同じ分散を持つ等方ガウス分布で構成されるGMMといえる.
  - マハラノビス距離を用いた場合は同じ分散を用いていないことになる.
  - コサイン類似度を用いると，混合von Mises Fisher分布を想定していることになる.



k-meansのクラスタリング結果



GMMのクラスタリング結果
所属クラスタがグラデーションになっている.

GMMとk-means

- GMMはデータが所属するクラスタを1つに絞らない.
  - 例えば，2つのクラスタに分ける場合，あるデータは一つのクラスタは0.8もう一つのクラスタは0.2ほど所属するというように，所属するかどうかを0か1ではなく連続値で表す.
  - これをsoft asignmentといい，soft asignmentを行うクラスタリングをsoft clustering (fuzzy clustering)という.
  - k-meansのようにクラスタに所属するかどうかを0か1で決めることをhard asignmentといい，このようなクラスタリングをhard clusterinという.