データサイエンス
第７回

~クラスター分析・クラス識別(1)~

情報理工学系研究科
創造情報学専攻
中山 英樹

本日の内容

- クラスター分析
  - 階層的クラスタリング
  - 非階層的クラスタリング
- クラス分類
  - パターン認識入門
  - ベイズの定理
  - 生成的アプローチ

クラスター分析の位置づけ

- 教師なし学習の一つ。特に、発見したい構造が離散的な場合

  - データから何かを発見したい → 教師なし学習

説明変数	手法
量的データ（比尺度）	主成分分析、因子分析、LPP
量的データ（間隔尺度）	クラスター分析、多次元尺度構成法、数量化Ⅳ類
質的データ	数量化Ⅲ類、対応分析

  - データを使って何かを予測したい → 教師あり学習

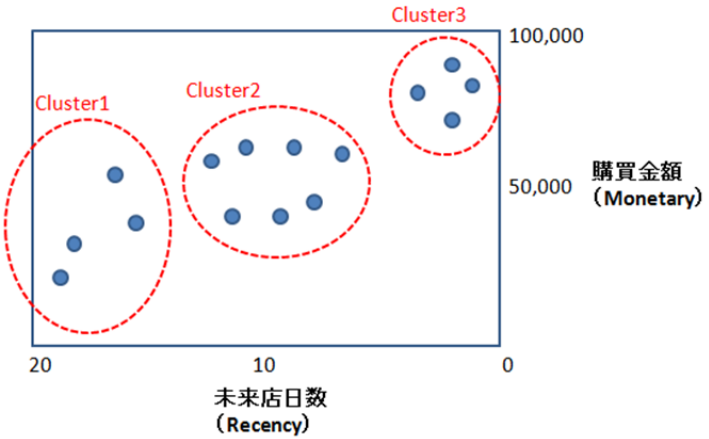
目的変数	説明変数	手法
量的データ	量的データ	回帰分析
	質的データ	数量化Ⅰ類
質的データ	量的データ	判別分析
	質的データ	数量化Ⅱ類

クラスター分析（クラスタリング）

- データ間の類似度（距離）を定義し、
  似ているものは同じまとまり（クラスタ）に、　← 内的結合
  似ていないものは異なるまとまりに分類する　← 外的分離
  - 雑多なデータの代表点を抽出したい
  - 購買データに基づくユーザのグループ分け、ウェブ上の文書分類、etc.
  - Cluster = ブドウの房

- 目的変数なし（教師なし）の多変量解析
  - 各クラスタの意味が自動的に
    得られるわけではない
  - あくまである一つの視点から見た場合

- 結果の解釈が重要
  - 各クラスタの意味は後で（人間が）解釈する



http://www.blog.rapidminer.jp/2012/03/rfm.html

距離の定義

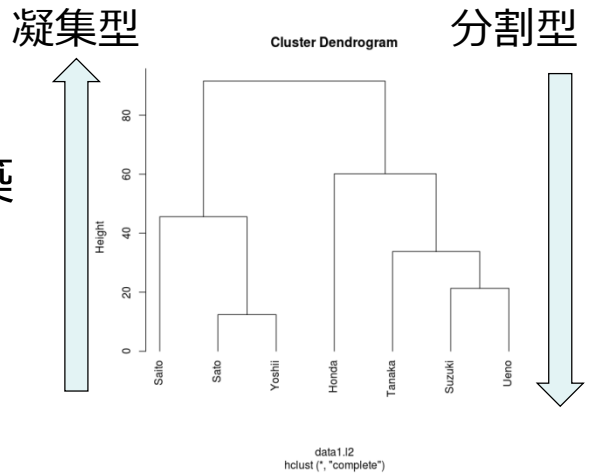
特徴ベクトル（説明変数）を $\mathbf{x} = \left(x_1, x_2, \ldots, x_d\right)^T$ と表記する

- ユークリッド距離（L2距離） $\left[\sum_{k=1}^{d}\left(x_{ik} - x_{jk}\right)^2\right]^{1/2}$
- ミンコフスキー距離（Lp距離） $\left[\sum_{k=1}^{d}\left|x_{ik} - x_{jk}\right|^p\right]^{1/p}$
- マハラノビス距離 $\left(\mathbf{x}_i - \overline{\mathbf{x}}\right)^T \Sigma^{-1}\left(\mathbf{x}_j - \overline{\mathbf{x}}\right)$

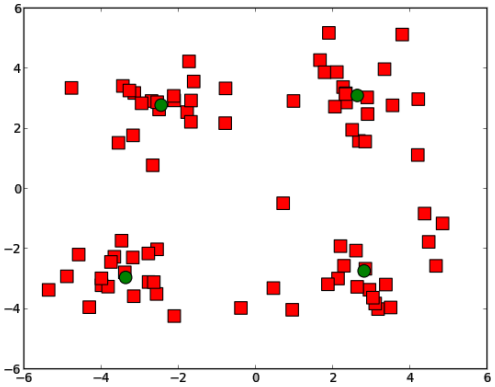
  $\overline{\mathbf{x}}, \Sigma$ はそれぞれサンプルの平均・共分散行列
  説明変数間に大きなスケールの差がある時に使う
- コサイン類似度 $\dfrac{\mathbf{x}_i^T \mathbf{x}_j}{\left\|\mathbf{x}_i\right\|\left\|\mathbf{x}_j\right\|}$
- …

クラスタリング手法の分類

- 階層的手法（凝集型）
  - データ一つ一つがクラスタである状態から出発し、ボトムアップに階層的な分類構造を構築
  - 実際はそれほど使われないかも
    - サンプル数が大きいと計算が大変＆解釈が難しい



- 非階層的手法
  - クラスタの良さを定義する目的関数を定義し、これを最適にする分割を探索する
  - 一般的にはこちらがよく使われる

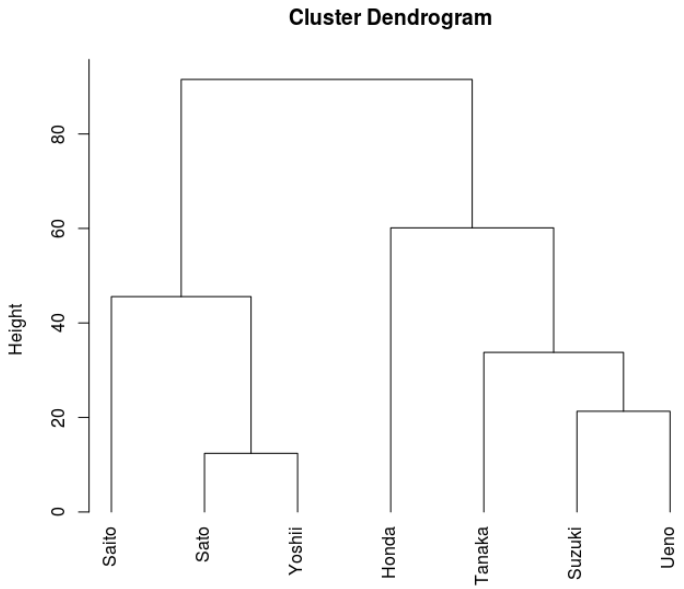


階層的クラスタリング

- １．各データを１つのクラスタとする
- ２．クラスタ間の距離（類似度）に基づいて
  クラスタを逐次的に併合する
- ３．１つのクラスタになるまで併合を繰り返す

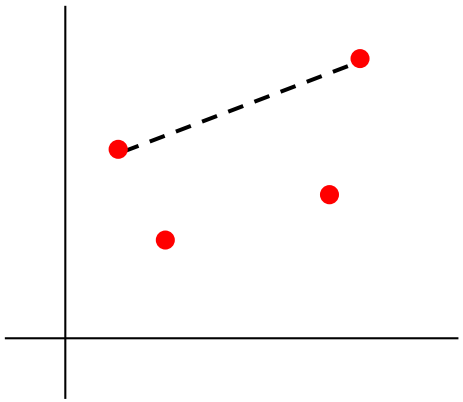
- データの階層構造を表す樹形図
  （デンドログラム）が得られる

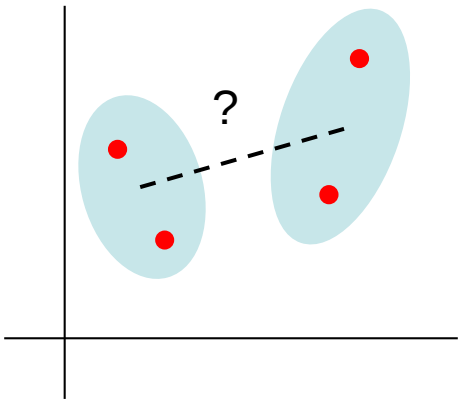
ここの高さが
そのクラスタ間の距離



階層的クラスタリングの手法

- 与えられているのはデータ間の類似度
- クラスタ間の距離の測り方はいろいろあり得る
  - 単連結法
  - 完全連結法
  - 群平均法
  - ウォード法
  - （重心法）
  - （メディアン法）

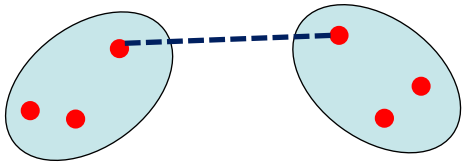




クラスタ間の距離の定義（１）

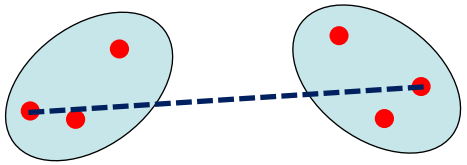
- 最短距離法（単連結法）
  - 二つのクラスターの個体のうち最も近い個体間の距離

$$d(C_1, C_2) = \min_{\mathbf{x}_1 \in C_1, \mathbf{x}_2 \in C_2} d(\mathbf{x}_1, \mathbf{x}_2)$$



- 最長距離法（完全連結法）
  - 二つのクラスターの個体のうち最も遠い個体間の距離

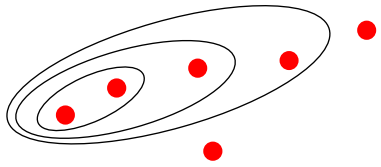
$$d(C_1, C_2) = \max_{\mathbf{x}_1 \in C_1, \mathbf{x}_2 \in C_2} d(\mathbf{x}_1, \mathbf{x}_2)$$



○ 計算コストが（比較的）小さい
（最小全域木）

× 外れ値の影響を受けやすい

× “チェイニング”が起こりやすい

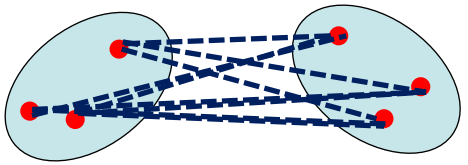


チェーン状のクラスタ

クラスター間の距離の定義（２）

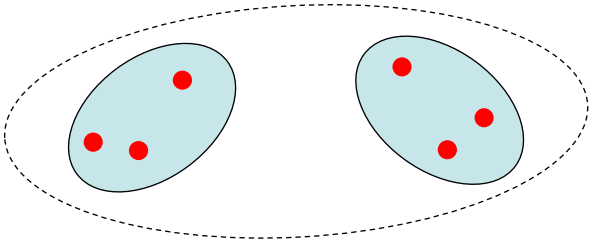
- 群平均法
  - 二つのクラスターのすべての個体間の距離の平均

$$d(C_1, C_2) = \frac{1}{|C_1||C_2|} \sum_{\mathbf{x}_1 \in C_1} \sum_{\mathbf{x}_2 \in C_2} d(\mathbf{x}_1, \mathbf{x}_2)$$



- ウォード法（最小分散法）
  - 結合後の分散と、結合前のクラスタそれぞれの分散の和との差
  - 判別規準と同じ

$$d(C_1, C_2) = \mathrm{var}(C_1 \cup C_2) - (\mathrm{var}(C_1) + \mathrm{var}(C_2))$$

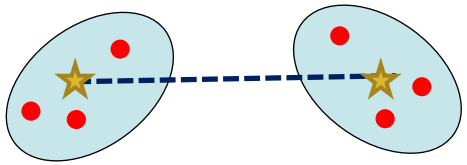


○外れ値に強く、実用性が高い

クラスター間の距離の定義（３）

- 重心法
  - 二つのクラスターの重心間の距離

$$d(C_1, C_2) = d\left(\frac{1}{|C_1|}\sum_{\mathbf{x}_1 \in C_1}\mathbf{x}_1, \frac{1}{|C_2|}\sum_{\mathbf{x}_2 \in C_2}\mathbf{x}_2\right)$$



- 重み付重心法（メディアン法）
  - 各クラスターの個体数の違いを考慮

(今はあまり使われない)

○ 計算コストが小さい

× クラスターの統合過程で、前段階よりも小さい距離となる”距離の逆転”が起こる場合がある

分析例

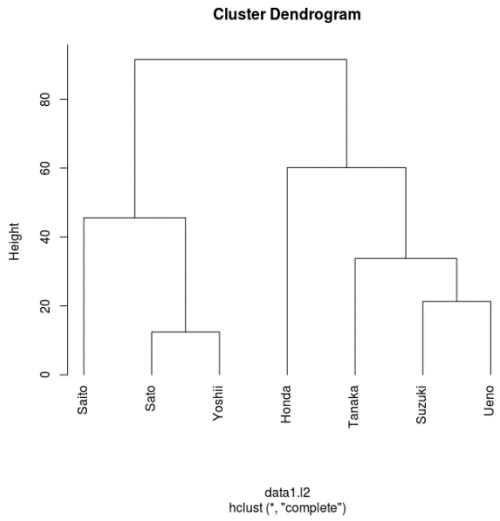
- まず距離行列を作る

```
> data1<-read.csv('seiseki.csv',header=F,row.names=1)
> data1.l2=dist(data1)　#ユークリッド距離
> data1.l2
          Tanaka     Sato       Suzuki     Honda     Ueno        Yoshii
Sato     68.65858
Suzuki   33.77869 81.11104
Honda    60.13319 64.14047 52.67827
Ueno     28.47806 60.75360 21.30728 47.10626
Yoshii   63.37192 12.40967 75.66373 54.31390 56.38262
Saito    67.88225 38.10512 87.53856 91.53142 67.72739 45.58509

> data1.l1=dist(data1,method="manhattan")　#L1距離
> data1.l1
      Tanaka   Sato  Suzuki  Honda  Ueno  Yoshii
Sato      144
Suzuki   47    173
Honda    102  132    95
Ueno     57    131    42       83
Yoshii    136   22    165      110     123
Saito     118   68    165      200     127     90
```

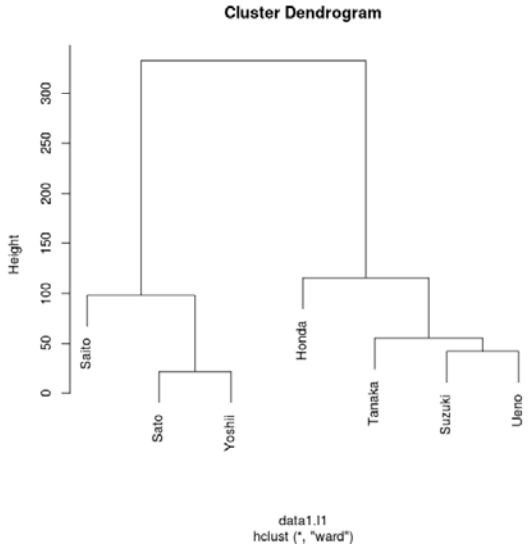
樹形図（デンドログラム）の表示

- 関数 hclust

```
> (data1.hc<-hclust(data1.l2))

Call:
hclust(d = data1.l2)

Cluster method   : complete
Distance         : euclidean
Number of objects: 7
> plot(data1.hc,hang=-1)
```



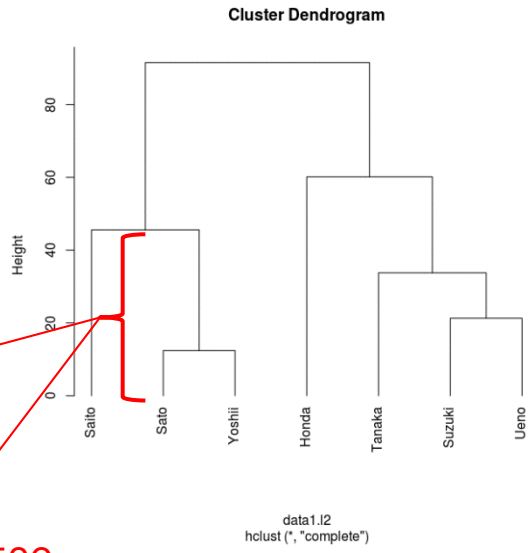

```
> (data1.hc<-hclust(data1.l1,method="ward.D"))

Call:
hclust(d = data1.l1, method = "ward")

Cluster method   : ward
Distance         : manhattan
Number of objects: 7
```



コーフェン行列

- 結合するタイミングでのクラスタ間距離を
  所属する各データの距離にとった行列
- デンドログラムはコーフェン行列に基づいて作成される
  - 高さがコーフェン行列の各要素に対応

```
> data1.hc<-hclust(data1.l2)
> cophenetic(data1.hc)
         Tanaka     Sato      Suzuki    Honda     Ueno      Yoshii
Sato     91.53142
Suzuki   33.77869 91.53142
Honda    60.13319 91.53142 60.13319
Ueno     33.77869 91.53142 21.30728 60.13319
Yoshii   91.53142 12.40967 91.53142 91.53142 91.53142
Saito    91.53142 45.58509 91.53142 91.53142 91.53142 45.58509
```



結果の妥当性

- コーフェン行列が元の距離行列と近いほどよしとする
  - 相関係数（コーフェン相関系数）

```
> cor(data1.l2,cophenetic(data1.hc))
[1] 0.8944869
```

  - できるだけコーフェン相関係数を保ちつつ
    クラスター数を絞っていく

- あくまで一つの指標
  - 分析結果が妥当であることを保証するものではない

非階層的クラスタリング

- データの分割の良さを表す評価関数を定義して、
  最適解を探索するアプローチ
- クラスタの数は、ハイパーパラメータとして与える手法が多い
  - 結果を解釈しながら、適当な数を探索
    （またはAIC,MDL等を用いて決める）
  - 気になる人はノンパラメトリックベイズを勉強しましょう

- いろんな手法
  - k-means
  - Fuzzy c-means
  - 混合分布モデル
  - スペクトラルクラスタリング
  - トピックモデル (pLSI, LDA)
  - NMF (Non-negative Matrix Factorization)
  - 自己組織化マップ
  - Mean-shift クラスタリング
  - …

k-means法

- 各クラスタに所属するサンプルの、クラスタ中心点までの距離の和が最小となるように配置

$$Err\left(\{C_k\}_{k=1}^{K}\right)=\sum_{k=1}^{K}\sum_{\mathbf{x}_i\in C_k} d\left(\overline{\mathbf{x}}_k,\mathbf{x}_i\right)$$

- クラスタ数*K*は与える
- 通常、距離 *d* はユークリッド距離の二乗（分散を最小化する）

- 実用上、非常に重要でよく用いられる手法
- c-means とも呼ばれていた

k-means法のアルゴリズム

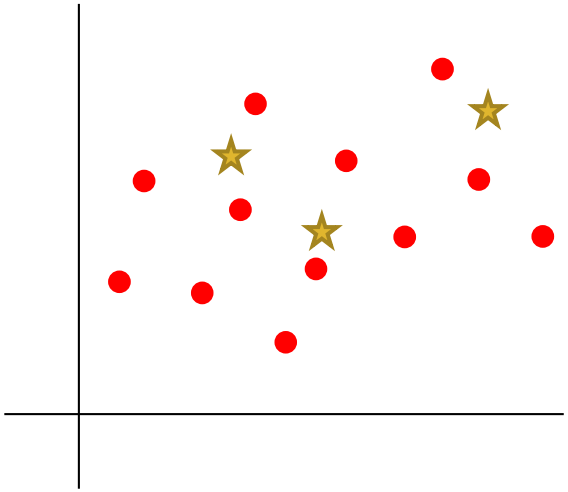
- １．全データ $\{\mathbf{x}_i\}_{i=1}^{N}$ からK個のデータをランダムに選び、
  クラスタ中心点 $\{V_j\}_{j=1}^{K}$ の初期値とする。

- ２．各データ $\mathbf{x}_i$と各クラスタの中心点 V_j との距離を求め、
  最も近い中心のクラスタに $\mathbf{x}_i$ を割り当てる。

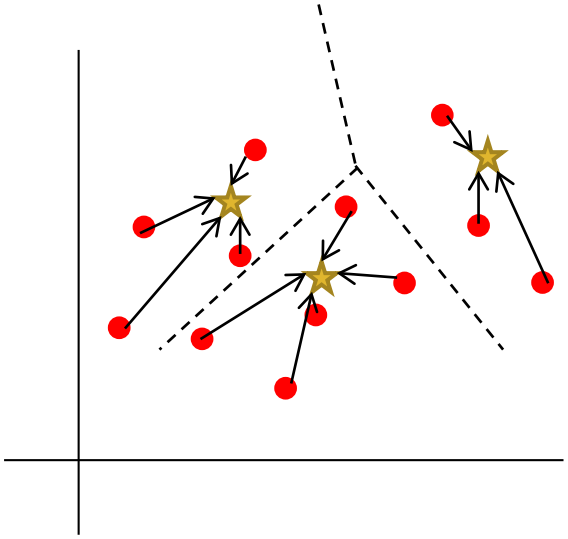
- ３．割り振ったデータをもとに各クラスタの中心 V_j を再計算する。
  計算は通常割り当てられたデータの各要素の算術平均が使用される。

- ４． ２～３を繰り返し、全ての $\mathbf{x}_i$のクラスタの割り当てが変化しなくなるか、目的関数の変化量が事前に設定した一定の閾値を下回った場合に、収束したと判断して処理を終了する。

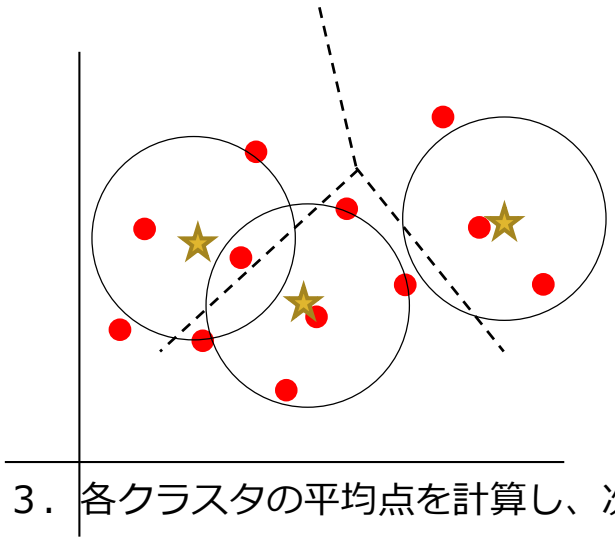
k-means法のアルゴリズム



1．ランダムにクラスタ中心を選択



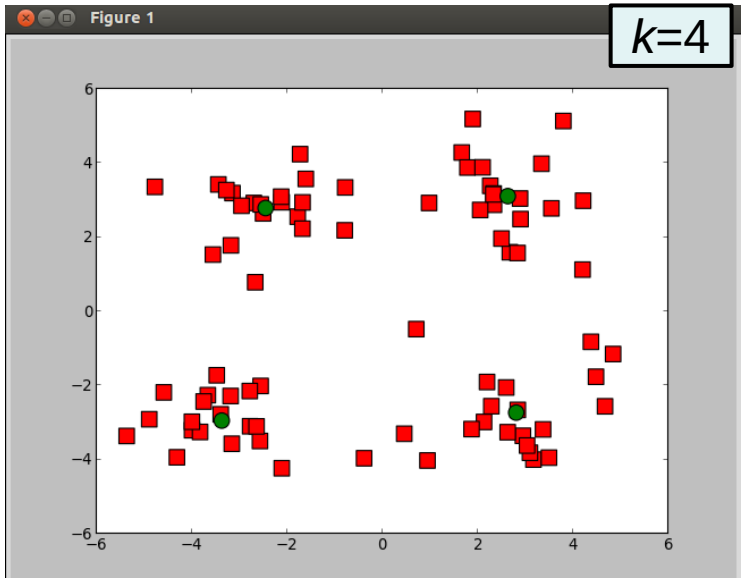
2．各データを中心が最も近いクラスタへ割り当てる

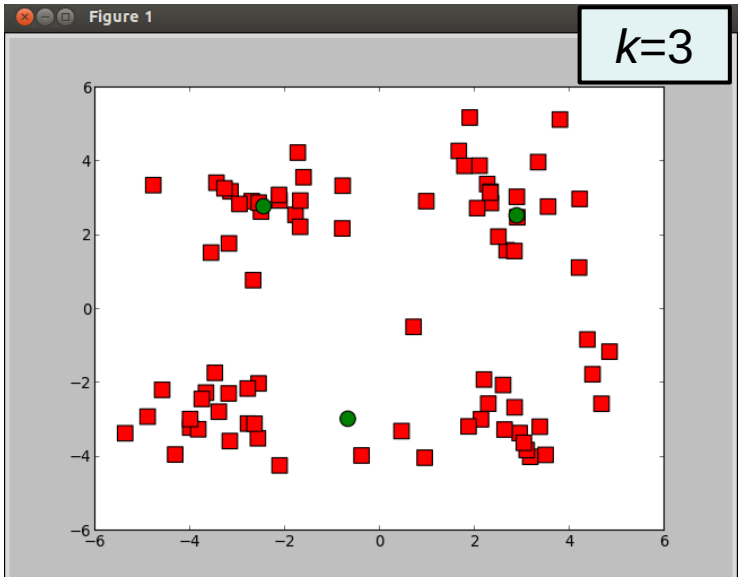


3．各クラスタの平均点を計算し、次の中心ベクトルとする

4．収束するまで繰り返し

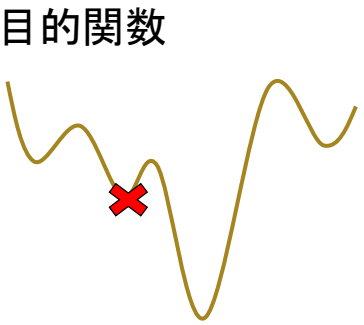
実行してみる (notebook)





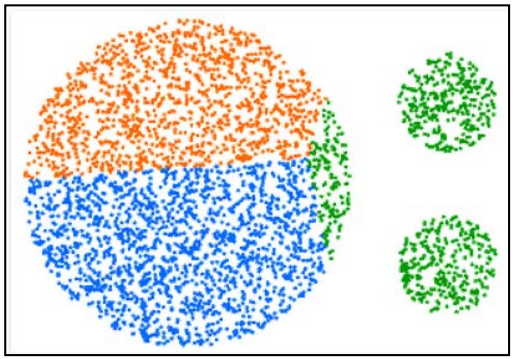
- この例だと３～４回のループで収束
- クラスタ数*k*を変えて何回か試してみる。
  いつも妥当な結果になるか？

k-means法の注意点



- k-meansの解は局所最適解しか保証されない
  - 初期値に強く依存
  - 通常、何度かk-meansを試行（毎回ランダムに*k*個の点を選びなおし初期値とする）し、目的関数が最小となった場合を採用する

- 分散の等しい超球状のクラスタを前提とする
  - 各クラスタに属するデータ数はおおむね等しいことを仮定
  - データとクラスタ数によっては必ずしも適切な結果にならない！



[Guha, 98]

k-means の改良手法（主に初期値依存性に対処）

- k-means++ [Arthur & Vassilvitskii, 2007]
  - 初期値の取り方を工夫した改良手法
  - なるべく離れるように最初のクラスタ中心を配置

- Bisecting k-means
  - データ全体を内包する一つの大きなクラスタからスタートし、階層的にk-means (k=2)を行い各クラスタを二分割していく方法
    - オリジナルのk-meansよりもよい解が得られる場合がある
    - 計算コストが小さい
    - 階層構造が得られる

  - 配布プログラムのbiKmeansを参照

おまけ：k-meansとPCA

- K-means Clustering via Principal Component Analysis
  [Ding & He, ICML2004]

- k=2の場合、k-meansの結果はPCAの
  最大固有値の軸（第一主成分）の正負と一致

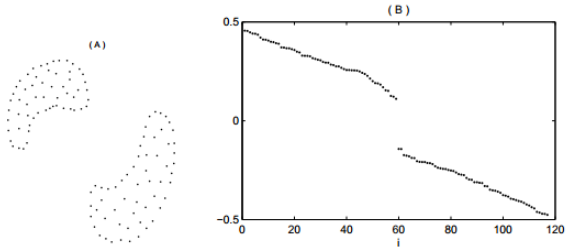


Figure 1. (A) Two clusters in 2D space. (B) Principal component $\mathbf{v}_1(i)$, showing the value of each element i.

- k>2の場合、k-meansで得られるk個のクラスタ中心ベクトルが張る空間が、PCAの上位(k-1)個の固有ベクトルが張る空間と一致
  - PCAでk-1次元へ圧縮してからk-meansをかけてもそれほど変わらない

- PCAは潜在的にクラスタリングをしているとも解釈できる
  - 連続値に緩和した分割問題を解いている
  - より一般には、スペクトラルクラスタリングと呼ばれる分野

Fuzzy c-means法

- k-means (=c-means) の改良版
- 各事例 $\mathbf{x}_i$ を複数のクラスタへ重みをつけて割り当てる

$$Err\left(\{u_{ki}\},\{\boldsymbol{\mu}_k\}_{k=1}^{C}\right)=\sum_{k=1}^{C}\sum_{i=1}^{N}\left(u_{ki}\right)^m d\left(\boldsymbol{\mu}_k,\mathbf{x}_i\right)$$

$$u_{ki}\in[0,1]\quad\sum_{k=1}^{C}u_{ki}=1\ \left(i=1,\ldots,N\right)$$

メンバシップ係数

m ：割り当てのファジーさを決めるパラメータ
（Cと共に与える）

- Fuzzy：曖昧な～

Fuzzy c-means法のアルゴリズム

- 1．全データのメンバシップ $\{u_{ki}\}$ をランダムに初期化
- 2．現在の $\{u_{ki}\}$ を用いて、各クラスタの中心点 $\boldsymbol{\mu}_k$ を計算

$$\boldsymbol{\mu}_k = \frac{\sum_{i=1}^{N} (u_{ki})^m \mathbf{x}_i}{\sum_{i=1}^{N} (u_{ki})^m}$$

- 3．$\{\boldsymbol{\mu}_k\}$ を用いて、データ $\mathbf{x}_i$ のクラスタkへの割り当てを更新

$$u_{ki} = \left[\sum_{j=1}^{C} \left(\frac{d(\mathbf{x}_i, \boldsymbol{\mu}_k)}{d(\mathbf{x}_i, \boldsymbol{\mu}_j)} \right)^{\frac{1}{m-1}} \right]^{-1}$$

- 4．２～３を繰り返し、パラメータの変化が十分小さくなれば処理を終了。

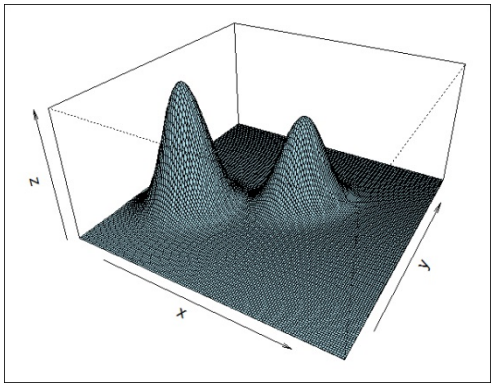
混合分布モデルを用いたクラスタリング

- データの背後に存在する確率モデルを推定し、クラスタリングへ利用する

- 混合正規分布 (Gaussian Mixture Model, GMM)
  - 複数の正規分布を足し合わせた確率モデル

$$p(\mathbf{x};\Theta)=\sum_{k=1}^{K}\alpha_k \mathrm{N}(\mathbf{x};\boldsymbol{\mu}_k,\Sigma_k) \quad \alpha_k\in[0,1] \quad \sum_{k=1}^{K}\alpha_k=1$$

$$\mathrm{N}(\mathbf{x};\boldsymbol{\mu}_k,\Sigma_k)=\frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\Sigma_k^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right)$$



  - クラスタリングのための手法ではないが、便利なので道具としてしばしばもちいられる。
  - 混合する正規分布の一つ一つをクラスタと解釈
  - k-means (Fuzzy c-means) はこれの特殊な場合

混合正規分布の最尤推定

- 対数尤度
  - 最大化したいが、少し厄介（ガウシアンの和の対数なので）

$$\log L(\Theta) = \sum_{i=1}^{N} \log \left\{ \sum_{k=1}^{K} \alpha_k \mathrm{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \Sigma_k) \right\}$$

  - 例えば…

$$\frac{\partial \log L(\Theta)}{\partial \boldsymbol{\mu}_k} = \frac{\partial}{\partial \boldsymbol{\mu}_k} \sum_{i=1}^{N} \log \left\{ \sum_{k=1}^{K} \alpha_k \mathrm{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \Sigma_k) \right\}$$

$$= \sum_{i=1}^{N} \left\{ \frac{\alpha_j \mathrm{N}(\mathbf{x}_i; \boldsymbol{\mu}_j, \Sigma_j)}{\sum_{k=1}^{K} \alpha_k \mathrm{N}(\mathbf{x}_i; \boldsymbol{\mu}_k, \Sigma_k)} \Sigma_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k) \right\}$$

$$= p(z_k = 1 | \mathbf{x}_i)$$

各データがk番目のガウシアンに所属する確率

EMアルゴリズム

- **観測不可能な潜在変数**に確率モデルが依存する場合に利用可能な最適化手法（最尤推定）

- Eステップ(expectation)、Mステップ(maximization)を交互に繰り返す
  - Eステップ：現在推定されているパラメータと潜在変数の分布に基づき、モデル尤度関数の条件付期待値を計算する
  - Mステップ：Eステップで求まった期待値を最大化するようにパラメータを更新する

- 対数尤度が単調増加することが保証されている
  - 一般に局所解へ収束する

GMM推定の手順

- 潜在変数=各データがどの正規分布に属しているか
  - クラスタリングにおいてはまさに求めたいもの
  - $\mathbf{x}_i$ がj番目の分布に属する確率を $q(i,j)$ と表記する（寄与率）

- 1．$\{\alpha_j^0, \boldsymbol{\mu}_j^0, \Sigma_j^0\}_{j=1}^K$ をランダムに初期化 (t=0)
- 2．Eステップ：現在のパラメータを固定し、潜在変数の分布を更新

$$\hat{q}(i,j)^{t+1} = \frac{\alpha_j^t \mathrm{N}(\mathbf{x}_i; \boldsymbol{\mu}_j^t, \Sigma_j^t)}{\sum_{k=1}^K \alpha_k^t \mathrm{N}(\mathbf{x}_i; \boldsymbol{\mu}_k^t, \Sigma_k^t)}$$

(尤度の期待値の導出は省略)

事後確率の比

- 3．Mステップ：更新されたqの分布に基づき、尤度を最大化

$$\alpha_j^{t+1} = \frac{\sum_{i=1}^N \hat{q}^{t+1}(i,j)}{N}$$

j番目の分布に属するデータ数の比

$$\boldsymbol{\mu}_j^{t+1} = \frac{\sum_{i=1}^N \hat{q}^{t+1}(i,j)\mathbf{x}_i}{\sum_{i=1}^N \hat{q}^{t+1}(i,j)} \qquad \Sigma_j^{t+1} = \frac{\sum_{i=1}^N \hat{q}^{t+1}(i,j)(\mathbf{x}_i - \boldsymbol{\mu}_j^{t+1})(\mathbf{x}_i - \boldsymbol{\mu}_j^{t+1})^T}{\sum_{i=1}^N \hat{q}^{t+1}(i,j)}$$

各データのj番目の分布への寄与で重みづけた平均と共分散行列

- 4．収束するまで２、３を繰り返し

GMM

- パラメータがとても多い
  - 計算が大変（収束が遅い）
  - 過学習しやすい
- 楽をするノウハウ
  - k-meansで初期化
  - 共分散行列を対角行列に限定
    …など

covariance_type :
{‘full’, ‘tied’, ‘diag’, ‘spherical’}

init_params :
{‘kmeans’, ‘random’}

class
sklearn.mixture.GaussianMixture(*n_components=1*, *covariance_type=’full’*, *tol=0.001*, *reg_covar=1e-06*, *max_iter=100*, *n_init=1*, *init_params=’kmeans’*, *weights_init=None*, *means_init=None*, *precisions_init=None*, *random_state=None*, *warm_start=False*, *verbose=0*, *verbose_interval=10*)

まとめ

- クラスタリング
  - 与えられたデータ集合をいくつかのまとまりに分割
  - 外的分離と内的結合
- 階層的クラスタリング（凝集型）
  - データ一つ一つがクラスタの状態から始めて，逐次的にクラスタを併合
  - 群平均法かウォード法
- 非階層的クラスタリング
  - クラスタの良さを定義する目的関数を最適にする分割を探索する
  - ハード（排他的なクラスタ）： k-means法
  - ソフト（複数クラスタへ重み付け）：混合正規分布など
- 注意すべきポイント
  - 適切な距離を使う
  - 得られたクラスタは絶対的・客観的なものではない
  - 自分で結果をみて解釈を与えることが重要

クラス分類（クラス識別）

- クラスタリングとは全く違うので注意

  - データから何かを発見したい → 教師なし学習

説明変数	手法
量的データ（比尺度）	主成分分析、因子分析、LPP
量的データ（間隔尺度）	クラスター分析、多次元尺度構成法、数量化Ⅳ類
質的データ	数量化Ⅲ類、対応分析

  - データを使って何かを予測したい → 教師あり学習

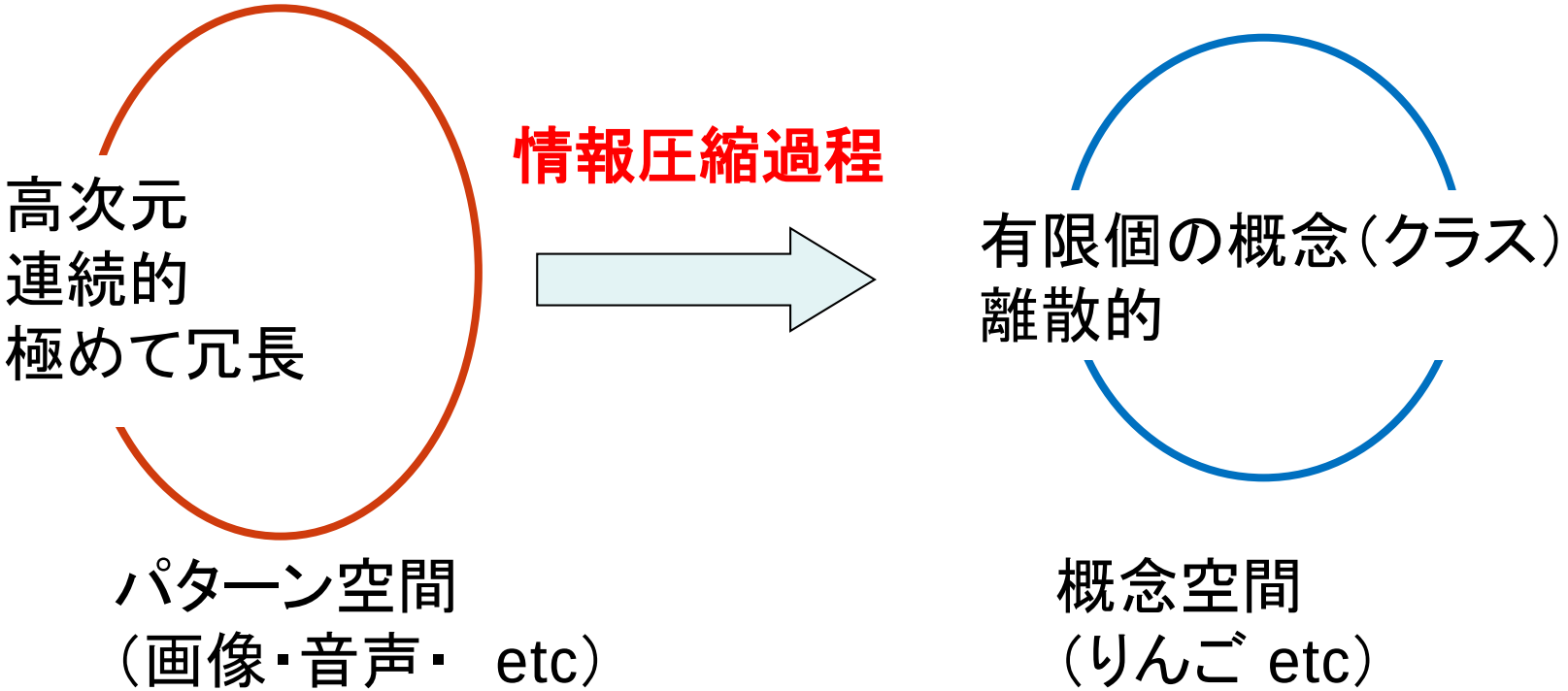
目的変数	説明変数	手法
量的データ	量的データ	回帰分析
	質的データ	数量化Ⅰ類
質的データ	量的データ	判別分析、SVM、kNN…
	質的データ	数量化Ⅱ類

本日の内容

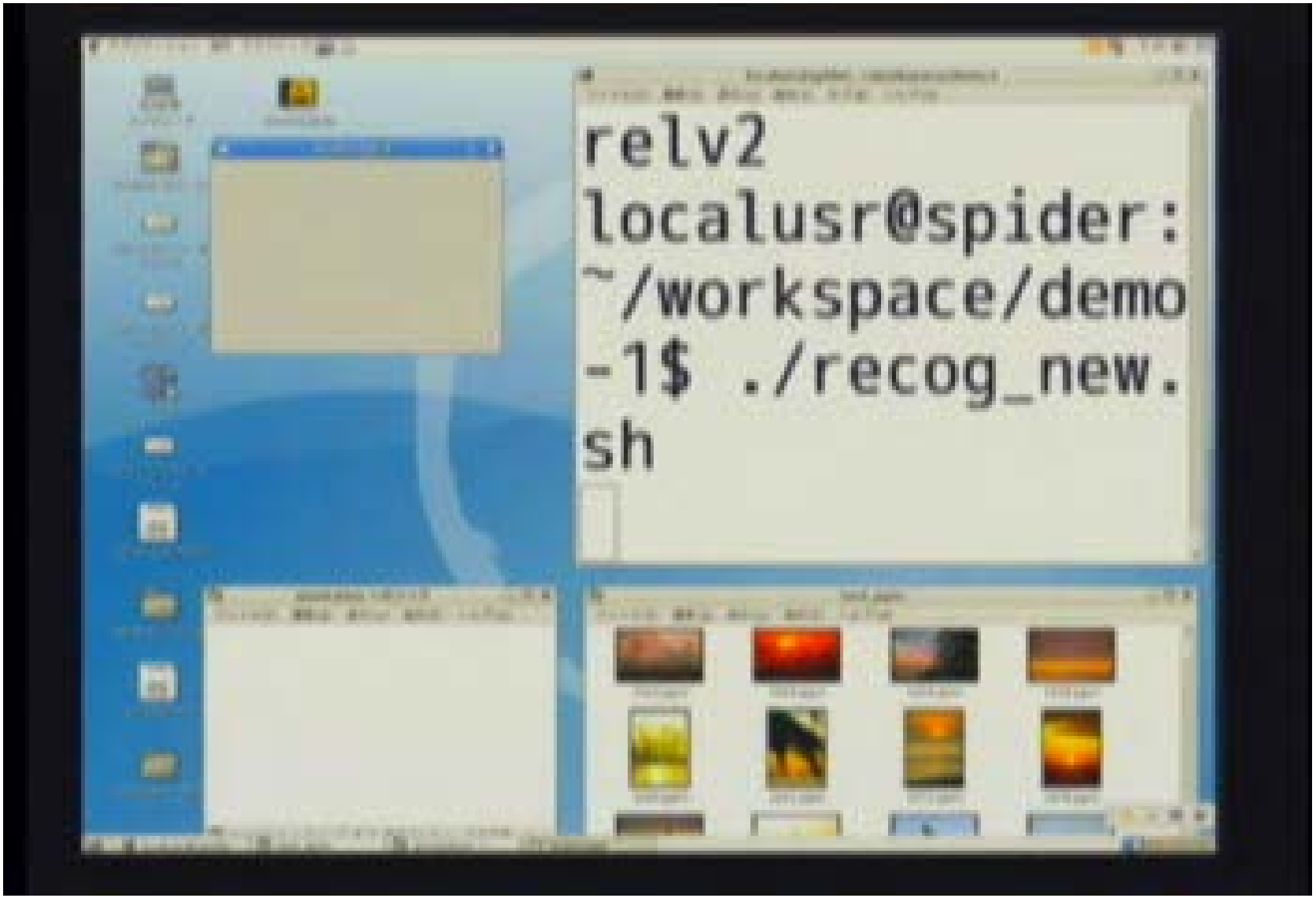
- クラス分類
  - パターン認識入門
  - ベイズの定理
- 各手法の外観・位置づけ
  - 識別的アプローチ
  - 生成的アプローチ
- 生成的アプローチの代表

応用：パターン認識

- 実世界の情報（＝パターン情報）を分類する（≒認識する）問題
- 最近は機械学習ありきになっている感があるが、もともとはその限りではない（ルールベースなど）



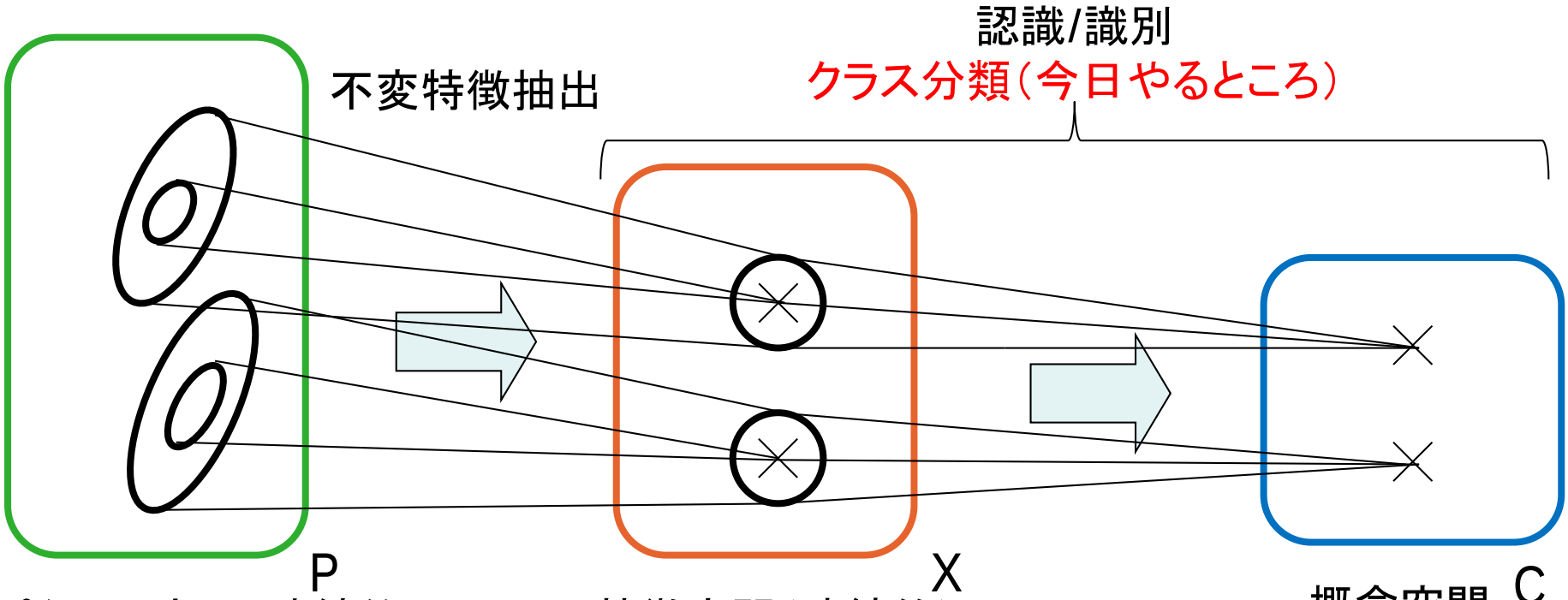
画像認識の例



[Fei-Fei et al. CVPR2007 Tutorial]

パターン認識のパイプライン

- よい特徴量（説明変数）を抽出・選択することは極めて重要
  - Garbage in garbage out
  - （ドメイン依存なので今日は扱いませんが…）



準備：ベイズの定理

Thomas Bayes (1702-1761)

- 条件付き確率（事後確率）：
  事象Aが起こったもとでBも起こる確率

$$P(B \mid A) = \frac{P(A,B)}{P(A)} = \frac{P(A \mid B)P(B)}{P(A)}$$

← Bの事前確率

Bの（Aに対する）事後確率

事後確率・事前確率を相互に書き換えることができる

- 覚え方

$$P(A,B) = P(B \mid A)P(A) = P(A \mid B)P(B)$$

パターン認識的には…

- パターンの特徴ベクトル x が与えられた時のクラスCの事後確率が重要

$$P(C \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid C)P(C)}{P(\mathbf{x})}$$

- 事後確率を最大とするクラスへ識別
  - 誤識別率を最小にする
  - 誤識別のリスク（ペナルティ）がクラスによらず一定の時、最適な識別境界を与える（ベイズ識別と一致）

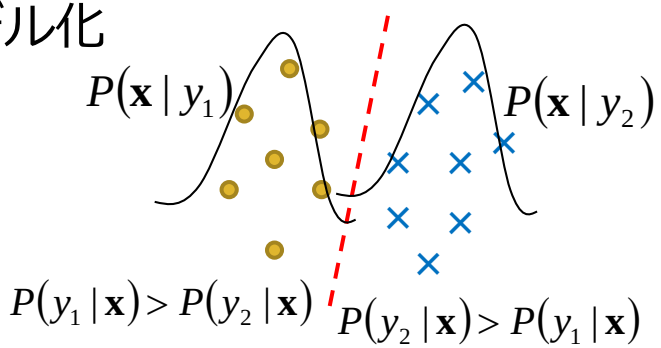
$$\hat{C} = \arg\max_{C} P(C \mid \mathbf{x})$$

分類のアプローチ（事後確率の推定方法）

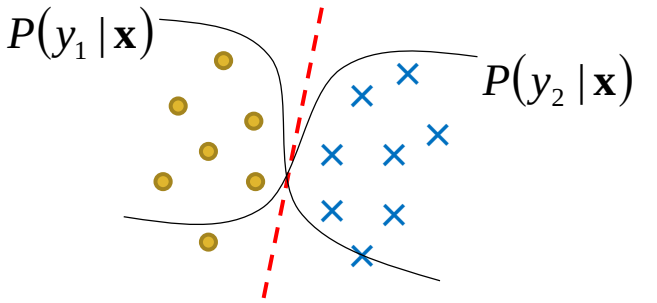
より一般的

- １．生成モデル
  - クラスごと条件付き確率と事前確率をモデル化
  - ナイーブベイズ
  - k-最近傍法

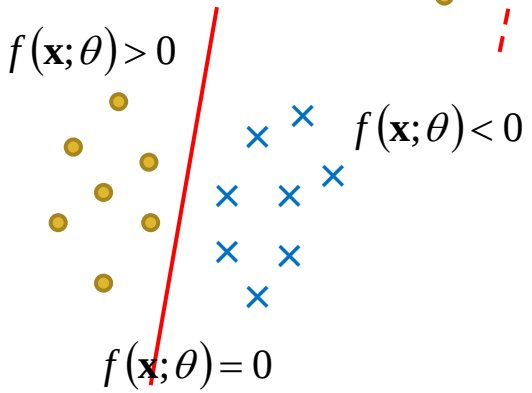
$$P(y \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid y)P(y)}{P(\mathbf{x})}$$



- ２．識別モデル
  - 事後確率を$P(y \mid \mathbf{x})$直接的にモデル化（元の分布はどうでもよい）
  - ロジスティック回帰



- ３．識別関数
  - 識別の境界面だけモデル化
  - SVM



より識別に特化

どちらのアプローチがよい？

- 一概には言えない
- 識別的アプローチのメリット
  - 閉じた世界（例えばベンチマークデータセット）における識別タスクでは性能が良いことが多い
    - 識別精度、計算コスト、メモリコスト、etc.
    - パラメータ数が比較的少なくて済む（→ 省サンプル、低計算コスト）

  「ある問題を解くとき、その途中段階で難しい問題を解かず、できるだけその問題を直接解くべきである」by V. Vapnik

- 生成的アプローチのメリット
  - タスクに関して何らかの事前知識を有している場合は特に有効
  - 未知のクラスに対処できる

生成的アプローチ

- 事後確率分布だけでなく、同時確率分布までモデル化

$$P(C \mid \mathbf{x}) \propto P(\mathbf{x} \mid C) P(C)$$

事後確率　条件付き確率　事前確率

  - クラスの事前確率は、何らかの先見知識がある場合を除き、単純にサンプルの割合で推定することが多い

$$\hat{P}(C) = \frac{N_C}{N}$$

- 条件付き確率（生成モデル）の推定がポイント
  例）正規分布を用い、最尤推定する（パラメトリック）

$$\hat{P}(\mathbf{x} \mid C) = N(\mathbf{x}; \hat{\mu}_C, \hat{\Sigma}_C)$$

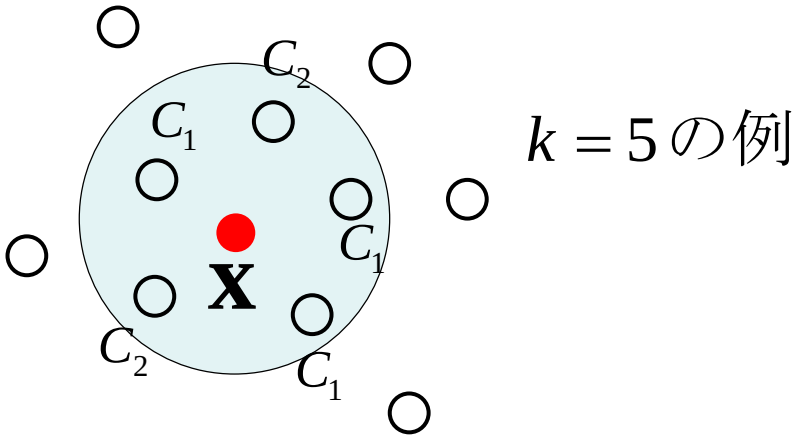
$\hat{\mu}_C, \hat{\Sigma}_C$ は最尤推定量、すなわちクラスCのサンプルの平均と分散

パラメトリックなモデルによる生成的分類

- 正規分布によるモデル化の場合
  - 各クラスの共分散パラメータを一定と仮定した場合、線形判別分析と同じ識別境界が得られる
- より複雑なモデル（GMMなど）は実際は扱いが難しい
  - パラメータが多い
  - 計算コストが膨大
  - 学習サンプルも大量に必要
  - 推定自体困難
  - …

K-最近傍法 (K-nearest neighbor, K-NN)

- 識別則は非常にシンプル
  - パターン入力 *x* について、 最も近い上位*K*個の学習データの中で、最も多い数のデータが所属するクラスへ *x* を識別



- 直感的には
  - 入力に類似している学習データの多数決
  - データが増えると精度は上がるが、識別のコストは非常に大きくなる

K-NN:確率的な解釈

- クラスC_kに属する学習データ数をN_k,
  全学習データ数を $N = \sum N_k$ とする
- 入力 x を中心とし、x の最近傍K点を含む超球の体積をVとする
- 超球に含まれるC_kのサンプル数をK_kとする

超球の内部（局所領域）
については以下のように近似できる

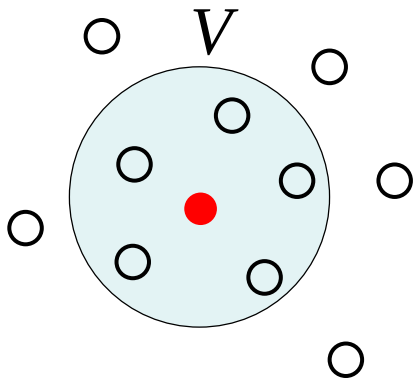
$$P(\mathbf{x}\,|\,C_k) = \frac{K_k}{N_k V}$$

$$P(\mathbf{x}) = \frac{K}{NV}$$

確率密度分布が局所的に
一定と近似

$$P(C_k) = \frac{N_k}{N}$$

$$\therefore P(C_k\,|\,\mathbf{x}) = \frac{P(\mathbf{x}\,|\,C_k)P(C_k)}{P(\mathbf{x})} \cong \frac{K_k}{K}$$

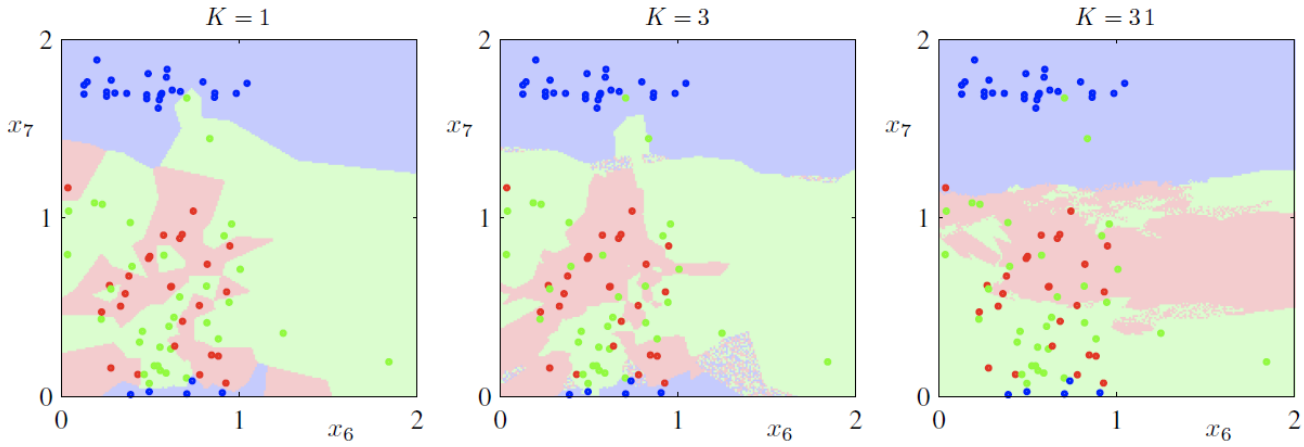


多クラス識別も自然に実現できる

K-NN: 確率的な解釈

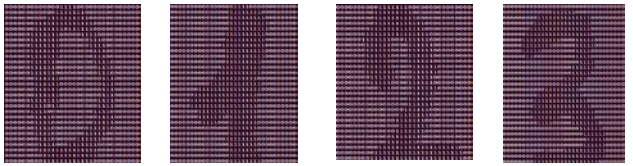
- 背後では、生成モデルの推定を行っていると解釈できる
  - 生成モデルのパラメータは置かないので、ノンパラメトリックな手法と呼ばれる
  - カーネル密度推定法と関連が深い

- Kは生成モデルの滑らかさを決定するハイパーパラメータ
  - モデルそのもののパラメータではないことがポイント
  - K→大：より大域的・単純な分布
  - K→小：より局所的・複雑な分布



手書き文字認識 (notebook)

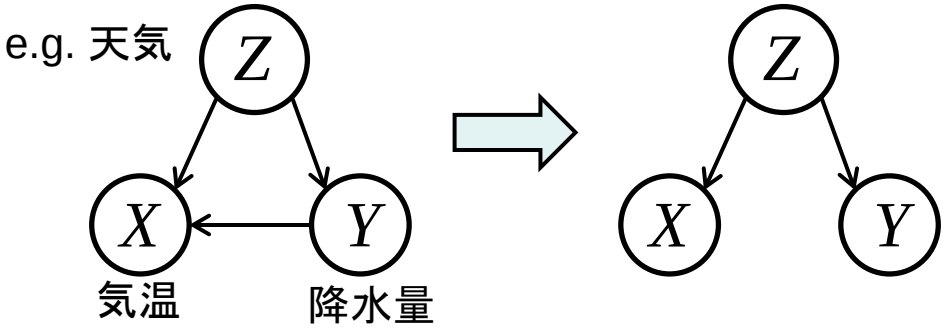
- 0~9の手書きの数字をKNNで認識してみる
  - 32x32サイズのバイナリ画像



ナイーブベイズ（単純ベイズ）

- もともと識別手法の名前ではない（非常に一般的かつ重要な概念）
  - 条件付きの同時確率を個々の条件付き確率の積へばらす近似方法

$$P(X,Y \mid Z) \quad \cong \quad P(X \mid Z)P(Y \mid Z)$$



  - Z が潜在的な構造をよく捉えていれば、比較的妥当な近似になると期待できる

ナイーブベイズ識別

- テキスト分類の基本的な手法
- ドキュメントを、出現する単語の集合として表現＝bag-of-words
  - 出現回数だけ利用
  - 各単語の位置や出現順などのコンテキストは考慮しない

ドキュメント

ナイーブベイズ識別

ドキュメント*D*の事後確率

$$P(C \mid D) \propto P(D \mid C) P(C)$$

訓練サンプル中の比率で近似（あるいは単に一定）

ナイーブベイズ

$$\hat{P}(D \mid C) = P(W_1, W_2, \ldots, W_n \mid C) \propto P(W_1 \mid C) P(W_2 \mid C) \cdots P(W_n \mid C)$$

$$P(W_i \mid C) = \frac{\text{カテゴリ}C\text{に属する訓練データ中の単語}W_i\text{の数}}{\text{カテゴリ}C\text{に属する訓練データの全単語数}}$$

- あらかじめ、訓練データ中の単語を数え上げておくだけで識別ができる！

例）スパムメール識別 (notebook)

非スパム

Hi Peter,

With Jose out of town, do you want to meet once in a while to keep things going and do some interesting stuff?

Let me know
Eugene

スパム

--- Codeine 15mg -- 30 for $203.70 -- VISA Only!!! --

-- Codeine (Methylmorphine) is a narcotic (opioid) pain reliever
-- We have 15mg & 30mg pills -- 30/15mg for $203.70 - 60/15mg for $385.80 - 90/15mg for $562.50 -- VISA Only!!! ---

実行結果

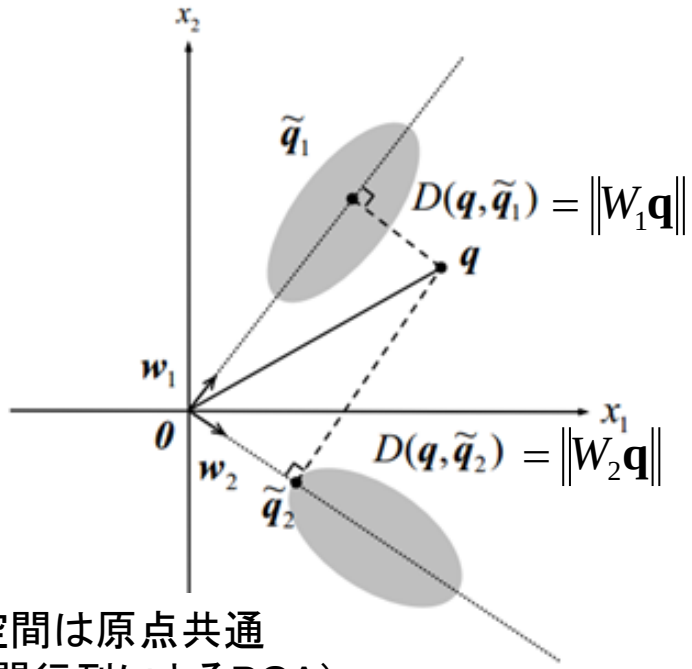
classification error ['home', 'based', 'business', 'opportunity', 'knocking', 'your', 'door', 'don', 'rude', 'and', 'let', 'this', 'chance', 'you', 'can', 'earn', 'great', 'income', 'and', 'find', 'your', 'financial', 'life', 'transformed', 'learn', 'more', 'here', 'your', 'success', 'work', 'from', 'home', 'finder', 'experts']

the error rate is: 0.1

(ランダムにサンプルを選ぶので毎回違った結果になる)

おまけ：部分空間法 (CLAFIC)

- 各クラスの成す部分空間 (PCAで学習）への近さを基準に識別
- 特徴量が線形な構造を有している場合に特に有効
- 日本発の技術



※部分空間は原点共通
（自己相関行列によるPCA）

距離最小基準：$\hat{C} = \arg\min_{C_i} \|W_i\mathbf{q}\|$

角度最小基準：$\hat{C} = \arg\min_{C_i} \dfrac{\|W_i\mathbf{q}\|}{\|\mathbf{q}\|}$

まとめ

- クラス分類（クラス識別）
  - 前提として、特徴抽出は重要
  - ベイズの定理、事後確率、ベイズ識別則
  - 識別的アプローチ、生成的アプローチの違い

- 識別的アプローチ：クラス間の“違い”だけ分かればよい
  - 線形識別関数：SVM、最小二乗識別
  - 線形識別モデル：ロジスティック回帰
- 生成的アプローチ：クラスの分布も知りたい
  - k-NN、ナイーブベイズ

- 次回：識別的アプローチ