

# デジタルメディア処理2

担当: 井尻 敬

1

## デジタルメディア処理2、2018（前期）

4/19 序論 : イントロダクション, テクスチャ合成  
4/26 特徴検出1 : テンプレートマッチング、コーナー・エッジ検出  
5/10 特徴検出2 : DoG特徴量、SIFT特徴量、ハフ変換  
5/17 領域分割 : 領域分割とは、閾値法、領域拡張法、動的輪郭モデル  
5/24 領域分割 : グラフカット、モーフォロジー処理、Marching cubes  
5/31 パターン認識基礎1: パターン認識概論、サポートベクタマシン  
6/07 パターン認識基礎2: ニューラルネットワーク、深層学習  
6/14 パターン認識基礎3: オートエンコーダ  
**6/21 筆記試験 (50点満点)**(n点以下の場合レポート出すかも)  
**6/28 プログラミング演習 1 (基礎的な課題40点, 発展的な課題 20点)**  
**7/05 プログラミング演習 2**  
**7/12 プログラミング演習 3**  
**7/19 プログラミング演習 4**  
**7/26 プログラミング演習 5**

## Contents

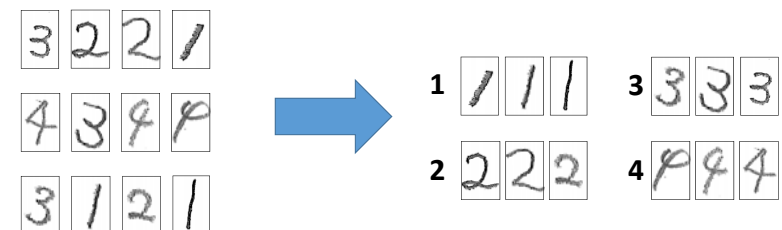
- パターン認識概論
- 特徴抽出
- 識別器
  - kNN
  - サポートベクターマシン
  - 決定木(Decision tree)とランダム森 (Random forests)

3

## パターン認識

『データの中の規則性を自動的に見つけ出し、その規則性を使ってデータを異なるカテゴリに分類する処理』 (PRML, C.M. Bishop)

例) 手書き文字画像の認識



4

## パターン認識

『データの中の規則性を自動的に見つけ出し、その規則性を使ってデータを異なるカテゴリに分類する処理』 (PRML, C.M. Bishop)

データ	研究分野
画像	画像認識 (Computer vision)
手書き文字	文字認識 (Optical character recognition)
音声	音声認識 (Speech recognition)
Genome	Bioinformatics
生体	Biometrics
:	:

5

## 身近な応用例 - 音声認識

iOS



siri

Windows



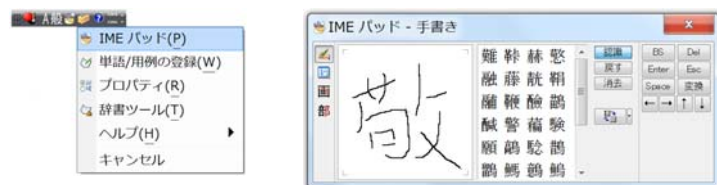
©Microsoft

Dictation

『コントロールパネル > 音声認識』

6

## 身近な応用例 - 文字認識



Windows IME pad

読めない漢字の手書きにより検索を支援

7

## 身近な応用例 - その他



指紋認証



顔認識



© IEEE Trans. Cyber. Hubert Shum, et al.

姿勢追跡  
ジェスチャ認識

8

# パターン認識

『データの中の規則性を自動的に見つけ出し、その規則性を使ってデータを異なるカテゴリに分類する処理』 (PRML, C.M. Bishop)

## 1) クラス分類 Classification

『複数の入力データを**既知**のクラスに分類する』

※クラス分類のみをパターン認識と呼ぶ事もある

## 2) クラスタリング Clustering

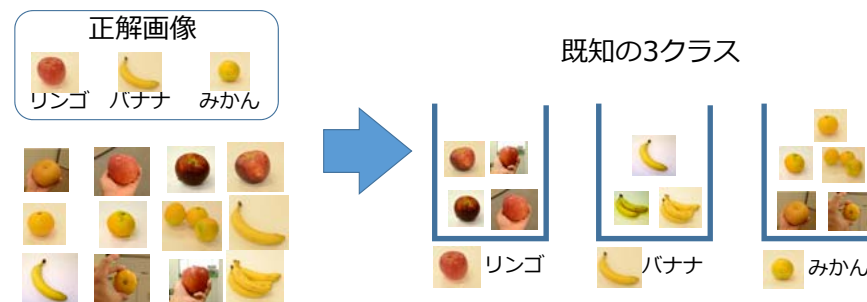
『複数の入力データから**未知**の類似したグループ (クラスタ) を発見する』

9

## 1) クラス分類 Classification

『複数の入力データを**既知**のクラスに分類する』

例) 果物の写真を、**リンゴ・バナナ・みかんの3クラス**に分類せよ

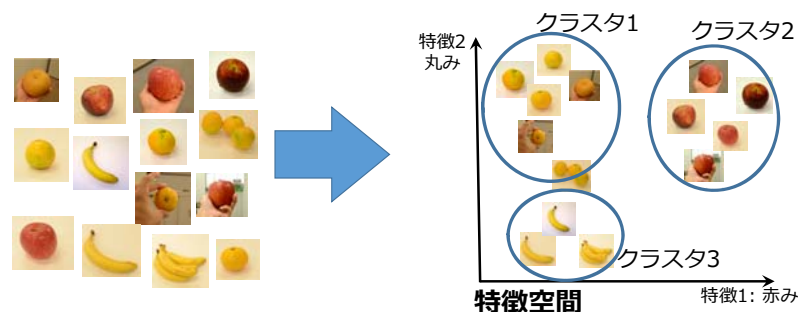


10

## 2) クラスタリング Clustering

『複数の入力データから**未知**の類似したグループ (クラスタ) を発見する』

例) 果物の写真を、**類似したグループ**を発見せよ



11

# パターン認識

『データの中の規則性を自動的に見つけ出し、その規則性を使ってデータを異なるカテゴリに分類する処理』 (PRML, C.M. Bishop)

## 1) クラス分類 Classification

『複数の入力データを既知のクラスに分類する』

## 2) クラスタリング Clustering

『複数の入力データから未知の類似したグループを発見する』

12

# パターン認識

『データの中の規則性を自動的に見つけ出し、その規則性を使ってデータを異なるカテゴリに分類する処理』 (PRML, C.M. Bishop)

## 1) クラス分類 Classification 本日の対象はこちら

『複数の入力データを**既知**のクラスに分類する』

※クラス分類のみをパターン認識と呼ぶ事もある

## 2) クラスタリング Clustering

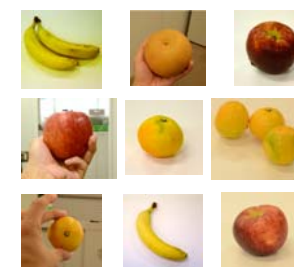
『複数の入力データから未知の類似したグループ(クラスタ)を発見する』

13

『写真を、リンゴ・バナナ・みかんの3クラスに分類せよ』



正解画像群  
クラスIDが既に付いた画像群  
(教師データと呼ばれる)

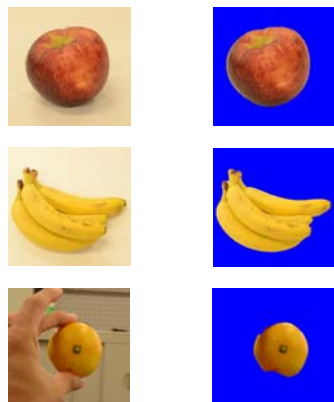


分類対象画像群  
この画像を分類したい

14

『写真を、リンゴ・バナナ・みかんの3クラスに分類せよ』

**前処理** : 画像から前景領域を抽出する



自動分割に関する既存手法は多いのでどれかを使う  
Otsu method,  
Grab cut,  
Saliency map + graph cut

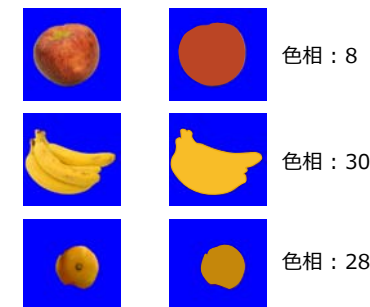
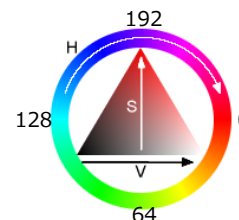
15

『写真を、リンゴ・バナナ・みかんの3クラスに分類せよ』

**特徴抽出** : 画像からクラスを良く分離する特徴量 (数値データ) を抽出する

### 1. 平均の彩度

- 前景領域の平均の色
- HSV色空間の色相H



16

## 『写真を、リンゴ・バナナ・みかんの3クラスに分類せよ』

**特徴抽出:** 画像からクラスを良く分離する特徴量（数値データ）を抽出する

2. 円形度：領域が円に近い度合

$$\frac{A}{L^2/4\pi}$$

A : 領域の面積  
L : 領域の周囲長  
 $L^2/4\pi$  : 周囲長Lの円の面積



円形度 1.0



円形度 0.785



円形度 0.604



円形度  
0.836



0.519



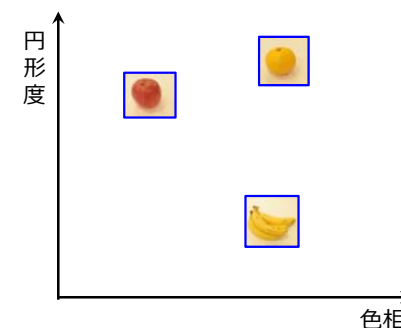
0.793

## 『写真を、リンゴ・バナナ・みかんの3クラスに分類せよ』

**特徴抽出:** 画像からクラスを良く分離する特徴量（数値データ）を抽出する

(1)平均色相と(2)円形度により、  
入力画像を**2D空間**に配置できる

特徴空間



## 『写真を、リンゴ・バナナ・みかんの3クラスに分類せよ』

**識別:** 特徴空間に入力画像を射影（配置）し、クラスIDを割り当てる

1. 正解画像を特徴空間に射影



ID: リンゴ

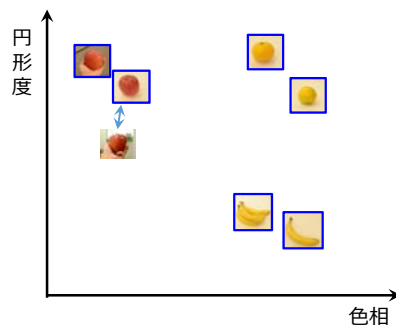
ID: バナナ

ID: みかん

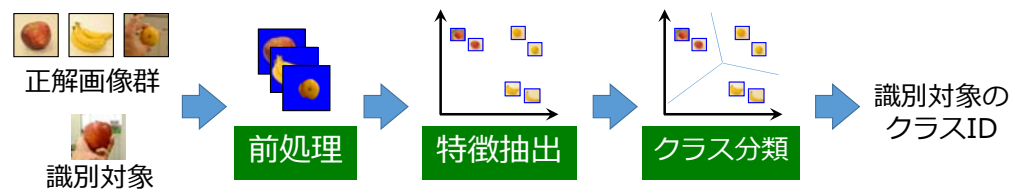
2. 分類したい画像も特徴空間射影し距離が一番近い正解画像のIDを返す



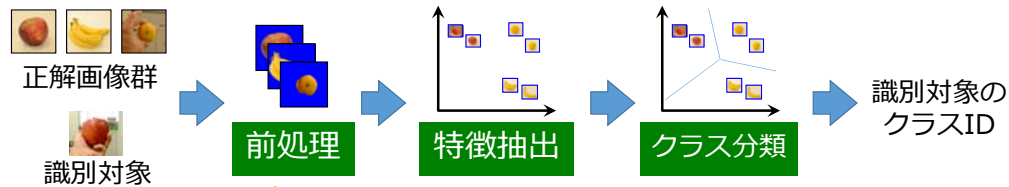
※ Nearest neighbor 法



## クラス分類の一般的な処理手順



## クラス分類の一般的な処理手順



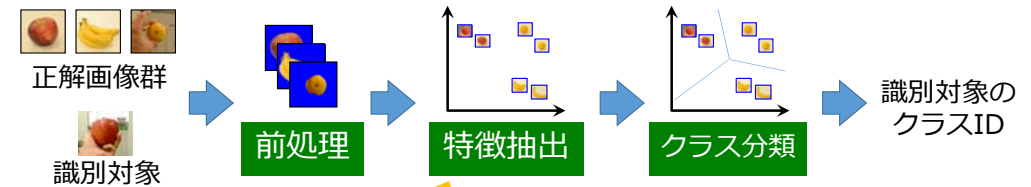
特徴抽出のための前処理

データが画像ならば…

二値化、平滑化、先鋭化、特徴保存平滑化、など

21

## クラス分類の一般的な処理手順



入力データ群に対し、同じクラスは近く・異なるクラス遠くなるような特徴空間にデータを射影する

良い特徴空間を構築するには、知識・経験・試行錯誤が必要

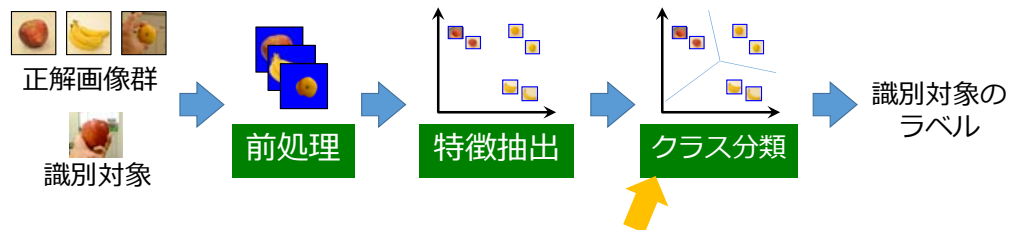
画像認識：HLAC・SIFT・HoG特徴などが有名

※最近流行りの深層学習は特徴量の設計もデータから学習する

※深層学習の発展に伴い、人がデザインした特徴量は「Hand Craftな」特徴量と呼ばれる

22

## クラス分類の一般的な処理手順



正解データ群を利用して特徴空間を分割する（訓練）

識別対象を特徴空間に射影し、上記の分割結果を用いてラベルを割り振る

### クラス分類の手法

K-Nearest Neighbor, ベイズ決定則, 決定木 (random forests), サポートベクタマシン  
ニューラルネットワーク, etc…

23

## まとめ：パターン認識とは

『データの中の規則性を自動的に見つけ出し、その規則性を使ってデータを異なるカテゴリに分類する処理』 (PRML, C.M. Bishop)

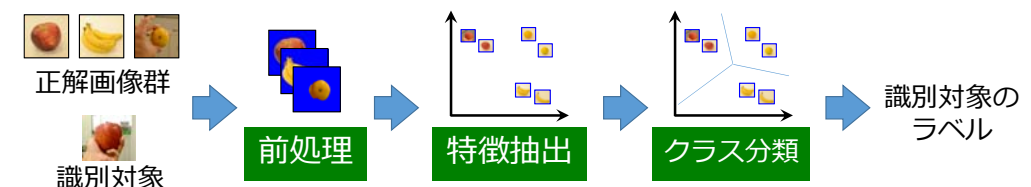
### 1) クラス分類 Classification

複数の入力データを既知のクラスに分類する  
※クラス分類のみをパターン認識と呼ぶ事もある

### 2) クラスタリング Clustering

複数の入力データから未知の類似したグループ（クラスタ）を発見する

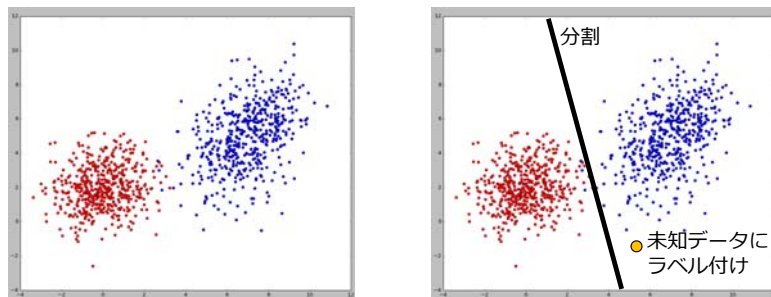
クラス分類の一般的な手順は以下の通り



24

## 識別器

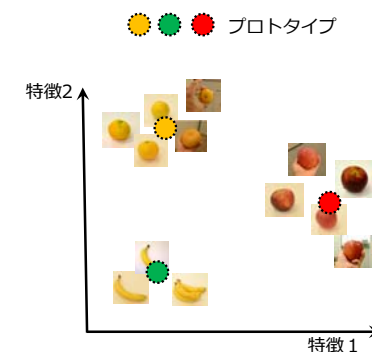
- 教師データ（ラベルつき特徴ベクトル）から特徴空間の分割方法を学習し、未知データにラベル付けを行なう手法
- プロトタイプ法, kNN (k-Nearest-Neighbor法), SVM (Support Vector Machine) RM (Random Forest)



25

## プロトタイプ法

- 各クラスを代表する点を選択（作成）  
↑これをプロトタイプと呼ぶ
  - 代表的なデータをプロトタイプにする
  - クラス内データの平均値をプロトタイプにする
- 未知データを特徴空間に配置し、最も近いプロトタイプのラベルを識別結果とする



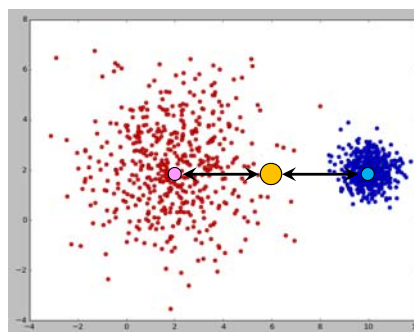
26

## プロトタイプ法 と マハラノビス距離

プロトタイプまでの距離で識別するのはOK  
でも明らかに分布の形が異なるクラス同士を  
ユークリッド距離で比較していいの？

右図において…

- 赤: 平均(2,2), 分散共分散  $\begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$  のガウス分布
- 青: 平均(10,2), 分散共分散  $\begin{pmatrix} 0.3 & 0 \\ 0 & 0.3 \end{pmatrix}$  のガウス分布
- 未知データ (6,2) はどちらのクラス？



27

## プロトタイプ法 と マハラノビス距離

N個の点群  $\mathbf{x}_i \in R^d$  の平均と分散共分散行列は…

$$\text{平均: } \mathbf{m} = \frac{1}{N} \sum_i \mathbf{x}_i$$

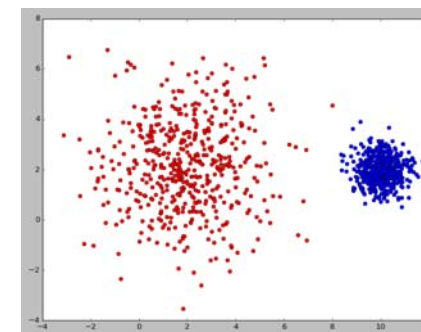
$$\text{分散共分散行列: } \mathbf{S} = \frac{1}{N} \sum_i (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T$$

点  $\mathbf{p} \in R^d$  と  $\mathbf{m}$  のユークリッド距離:

$$d = \sqrt{(\mathbf{p} - \mathbf{m})^T (\mathbf{p} - \mathbf{m})}$$

点  $\mathbf{p} \in R^d$  と  $\mathbf{m}$  のマハラノビス距離:

$$d = \sqrt{(\mathbf{p} - \mathbf{m})^T \mathbf{S}^{-1} (\mathbf{p} - \mathbf{m})}$$

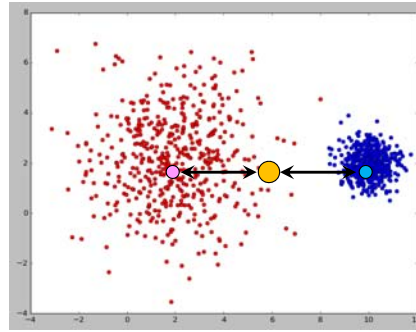


28



## プロトタイプ法 と マハラノビス距離

- 赤:平均(2,2), 分散共分散 $\begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$  のガウス分布
- 青:平均(10,2),分散共分散 $\begin{pmatrix} 0.3 & 0 \\ 0 & 0.3 \end{pmatrix}$  のガウス分布
- マハラノビス距離を用いた場合未知データ (6,2)はどちらのクラス?

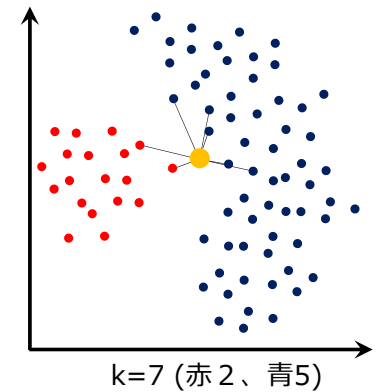


※マハラノビス距離は点群の分布を考慮し、分散の大きさの逆数で正規化した距離と考えられる

29

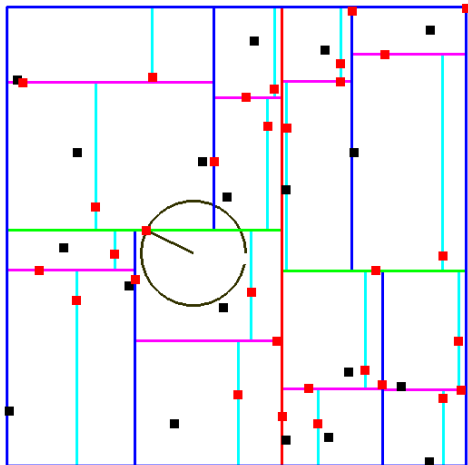
## kNN(k-Nearest Neighbor法)

- 特徴空間において、未知データとの距離が最も近いk個の教師データを検索し、その点の多数決でラベルを決定する
- 特徴空間の次元が低く教師データの量が十分多いときには高い精度が得られる
- 全教師データを保持するのでメモリ消費が大きい
- 素朴な実装をすると計算量も大きくなる



30

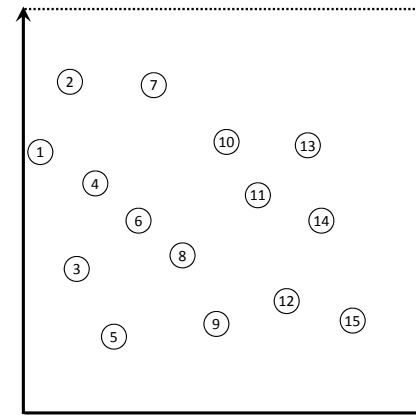
## kd-tree



- K-dimensional tree
- 2分木構造により空間を分割し、高速な近傍探索を可能にする
- 近傍探索の計算複雑度は  
平均  $O(\log N)$   
最悪ケース  $O(N)$

31

## kd-treeの構築



- 下を繰り返す

空間を分割する軸を決定し軸に沿って点群をソート

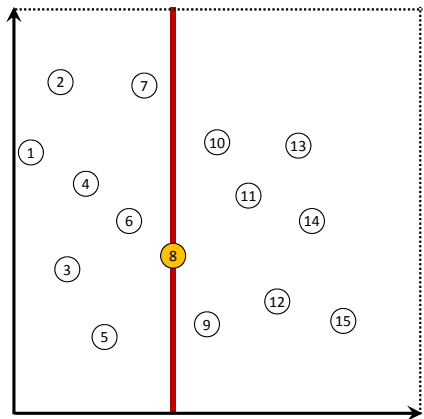
中央の点を現在ノードに割り当て、左側の点群を左の子に、右側の点群を右の子にする

1,2,3,4,5,6,7,8,9,10,11,12,13,14,15

32



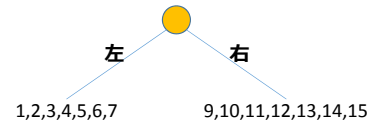
## kd-treeの構築



### • 下を繰り返す

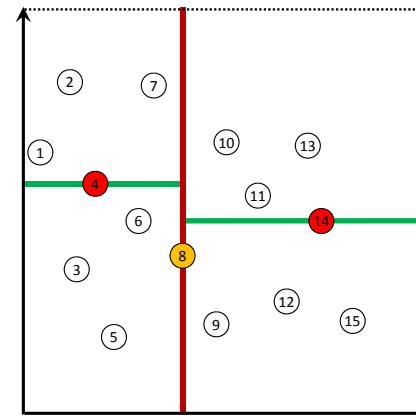
空間を分割する軸を決定し軸に沿って点群をソート

中央の点を現在ノードに割り当て、左側の点群を左の子に、右側の点群を右の子にする



33

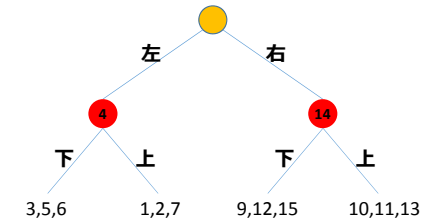
## kd-treeの構築



### • 下を繰り返す

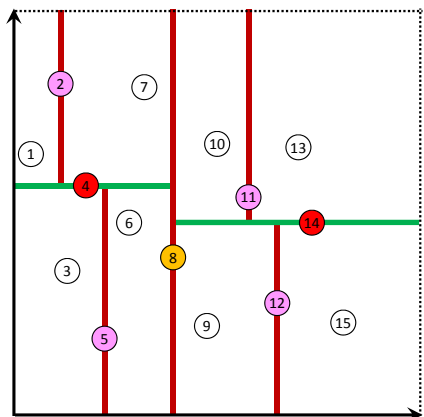
空間を分割する軸を決定し軸に沿って点群をソート

中央の点を現在ノードに割り当て、左側の点群を左の子に、右側の点群を右の子にする



34

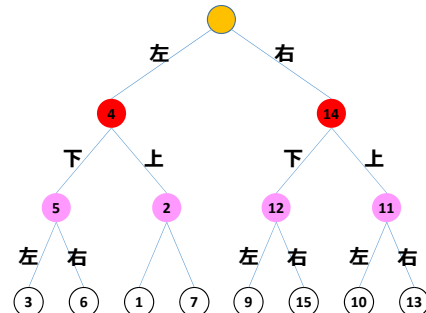
## kd-treeの構築



### • 下を繰り返す

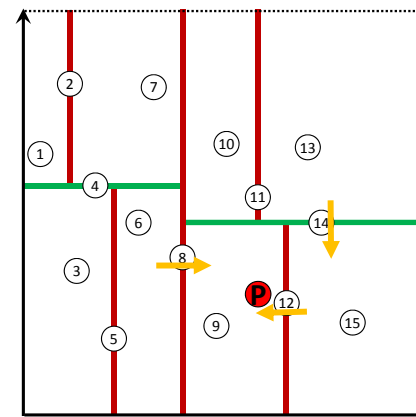
空間を分割する軸を決定し軸に沿って点群をソート

中央の点を現在ノードに割り当て、左側の点群を左の子に、右側の点群を右の子にする



35

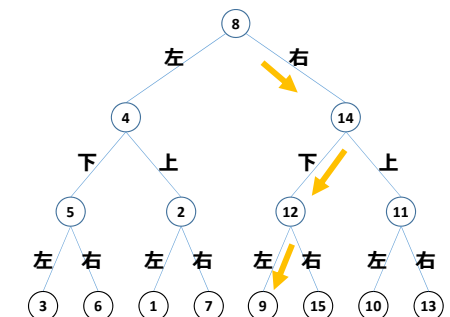
## kd-treeの構築



## 点pの最近傍点探索

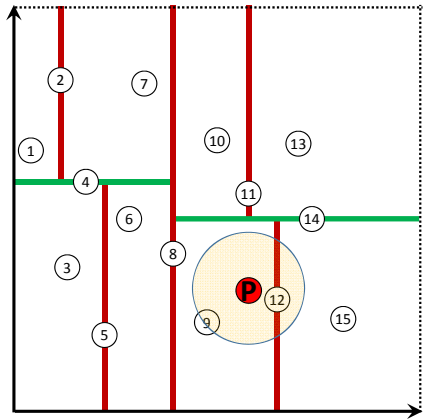
木を下方向にたどり葉ノードを見つけ、これを暫定的な最近傍点とする（近似解でよければここで終了）

到達した葉ノードから木を上方向にたどり、点pからの距離がR以下の領域は検索する、



36

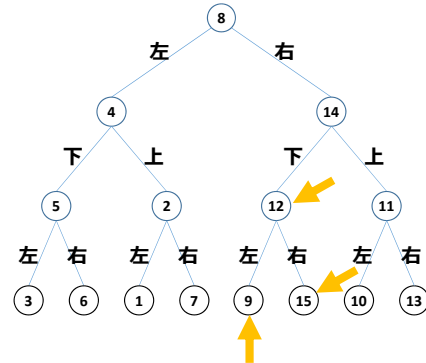
## kd-treeの構築



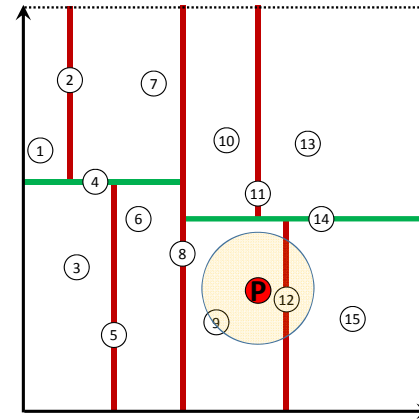
### 点pの最近傍点探索

木を下方方向にたどり葉ノードを見つけ、これを暫定的な最近傍点とする（近似解でよければここで終了）

到達した葉ノードから木を上方向にたどり、点pからの距離がR以下の領域は検索する、



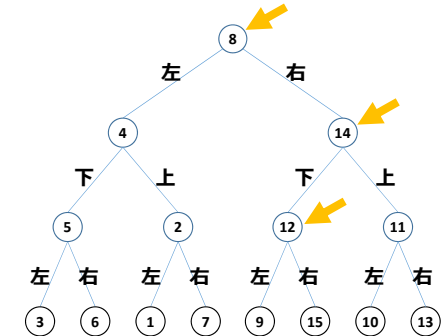
## kd-treeの構築



### 点pの最近傍点探索

木を下方方向にたどり葉ノードを見つけ、これを暫定的な最近傍点とする（近似解でよければここで終了）

到達した葉ノードから木を上方向にたどり、点pからの距離がR以下の領域は検索する、



## Decision tree と Random forests

## 決定木 (classification tree / decision tree)

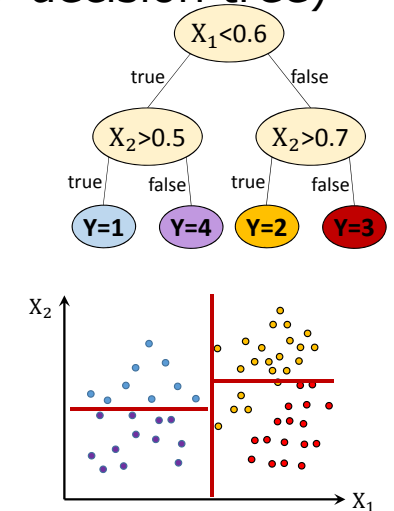
### 二分木でクラス分類を表現

**Node** : 特徴ベクトルに基づいた分割規則が定義される

**Leaf** : クラスに対応

• Xが観察される → 木を辿り分類先を決定

- 分類 (test) が高速
- 実装が簡単
- 木が深くなると過学習
- DNNに近い成績 (要発展)



## 決定木 (classification tree / decision tree)

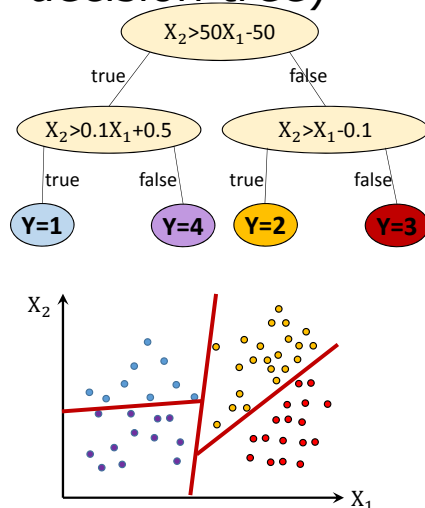
### 二分木でクラス分類を表現

**Node** : 特徴ベクトルに基づいた分割規則が定義される

**Leaf** : クラスに対応

•  $X$ が観察される  $\rightarrow$  木を辿り分類先を決定

- 分類 (test) が高速
- 実装が簡単
- 木が深くなると過学習
- DNNに近い成績 (要出展)

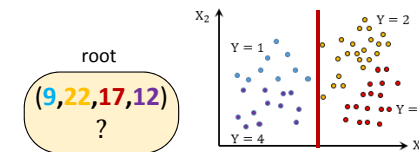


41

## 決定木の学習 (概要) [Fielding 77; Quinlan 93; Breiman 84]

入力: 教師データ  $(Y_i, X_i)$ , 木の深さ  $D$

1. Root に全教師データを関連付け
2. 深さが  $D$  になるまで以下を繰り返す
  - + Node  $d$  に注目
  - +  $d$  に属すデータ群を二分割するルールを決定
    - ランダムに候補を作成
    - **なるべく偏りが大きなルールを選択**
  - +  $d$  の子に分割したデータ群を関連付け
3. 葉にラベル付け (属するデータの多数決)

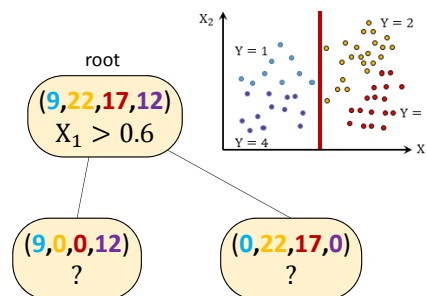


42

## 決定木の学習 (概要) [Fielding 77; Quinlan 93; Breiman 84]

入力: 教師データ  $(Y_i, X_i)$ , 木の深さ  $D$

1. Root に全教師データを関連付け
2. 深さが  $D$  になるまで以下を繰り返す
  - + Node  $d$  に注目
  - +  $d$  に属すデータ群を二分割するルールを決定
    - ランダムに候補を作成
    - **なるべく偏りが大きなルールを選択**
  - +  $d$  の子に分割したデータ群を関連付け
3. 葉にラベル付け (属するデータの多数決)

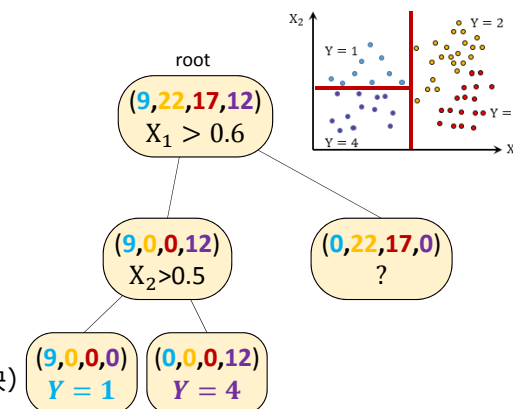


43

## 決定木の学習 (概要) [Fielding 77; Quinlan 93; Breiman 84]

入力: 教師データ  $(Y_i, X_i)$ , 木の深さ  $D$

1. Root に全教師データを関連付け
2. 深さが  $D$  になるまで以下を繰り返す
  - + Node  $d$  に注目
  - +  $d$  に属すデータ群を二分割するルールを決定
    - ランダムに候補を作成
    - **なるべく偏りが大きなルールを選択**
  - +  $d$  の子に分割したデータ群を関連付け
3. 葉にラベル付け (属するデータの多数決)

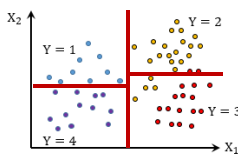
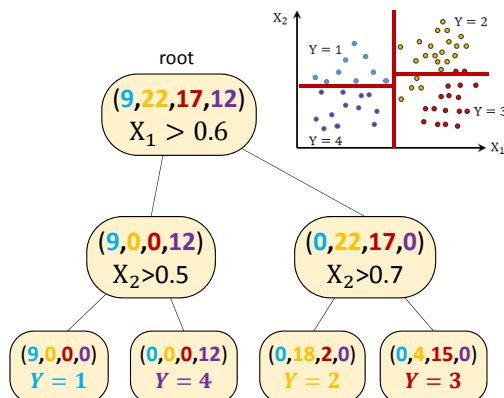


44

## 決定木の学習 (概要) [Fielding 77; Quinlan 93; Breiman 84]

入力：教師データ  $(Y_i, X_i)$ , 木の深さ  $D$

1. Root に全教師データを関連付け
2. 深さが  $D$  になるまで以下を繰り返す
  - + Node  $d$  に注目
  - +  $d$  に属すデータ群を二分割するルールを決定
    - ランダムに候補を作成
    - **なるべく偏りが大きなルールを選択**
  - +  $d$  の子に分割したデータ群を関連付け
3. 葉にラベル付け (属するデータの多数決)



## 参考資料

なるべく偏りが大きなルールを選択

例) 情報利得が大きくなる分割を選択

$$\text{Entropy: } H = -\sum_{c=1}^k P_c \log P_c$$

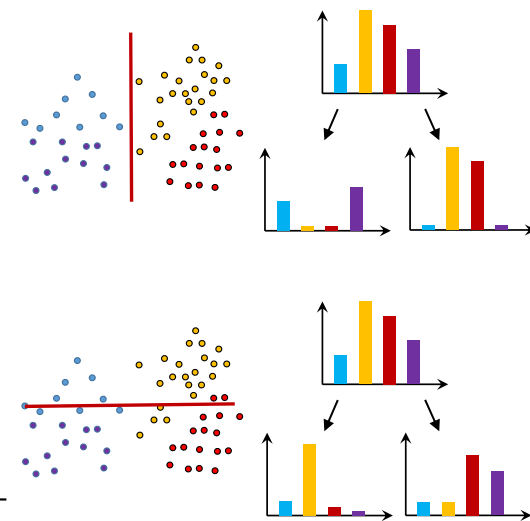
$P_c$  はクラス  $c$  に属すデータ点の出現確率

$$\text{情報利得: } H_p - \frac{|N_L|}{|N_p|} H_L - \frac{|N_R|}{|N_p|} H_R$$

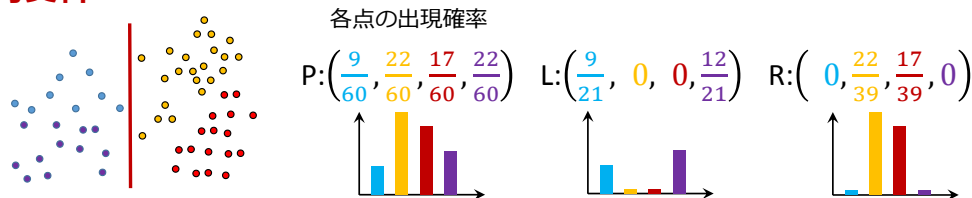
分割により減少したエントロピー量

$H_p/H_L/H_R$  : 親/左/右Nodeのエントロピー

$N_p/N_L/N_R$  : 親/左/右Nodeに属す要素数



## 参考資料



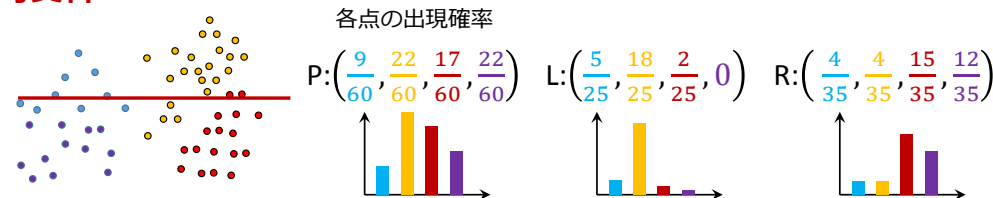
$$H_p = -\frac{9}{60} \log \frac{9}{60} - \frac{22}{60} \log \frac{22}{60} - \frac{17}{60} \log \frac{17}{60} - \frac{12}{60} \log \frac{12}{60} = 0.578$$

$$H_L = -\frac{9}{21} \log \frac{9}{21} - 0 - 0 - \frac{12}{21} \log \frac{12}{21} = 0.296$$

$$H_R = -0 - \frac{22}{39} \log \frac{22}{39} - \frac{17}{39} \log \frac{17}{39} - 0 = 0.297$$

$$\text{情報利得: } 0.578 - \frac{21}{60} 0.296 - \frac{39}{60} 0.297 = \mathbf{0.281}$$

## 参考資料



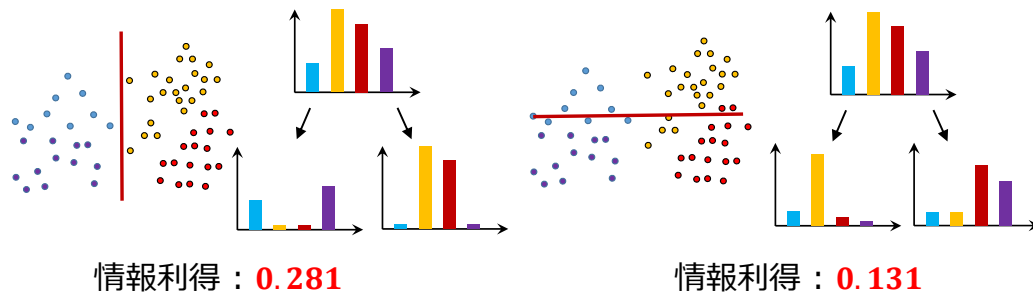
$$H_p = -\frac{9}{60} \log \frac{9}{60} - \frac{22}{60} \log \frac{22}{60} - \frac{17}{60} \log \frac{17}{60} - \frac{12}{60} \log \frac{12}{60} = 0.578$$

$$H_L = -\frac{5}{25} \log \frac{5}{25} - \frac{18}{25} \log \frac{18}{25} - \frac{2}{25} \log \frac{2}{25} = 0.330$$

$$H_R = -\frac{4}{35} \log \frac{4}{35} - \frac{4}{35} \log \frac{4}{35} - \frac{15}{35} \log \frac{15}{35} - \frac{12}{35} \log \frac{12}{35} = 0.532$$

$$\text{情報利得: } 0.578 - \frac{25}{60} 0.330 - \frac{35}{60} 0.532 = \mathbf{0.131}$$

## 参考資料



左の分割のほうが情報利得が高い（偏りが大きい）  
→ この二つの候補があったら左を選ぶ

49

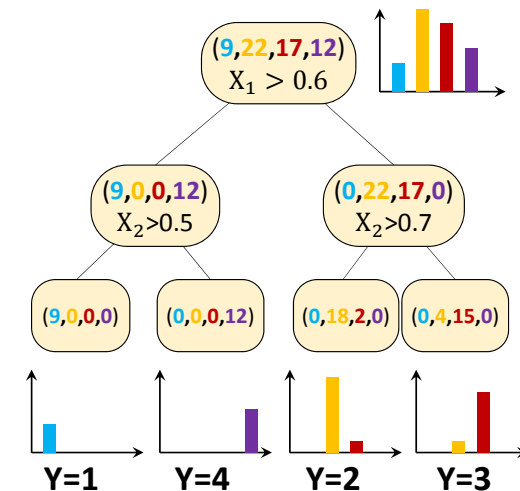
## 参考資料

### 葉にラベル付け

Nodeの分割を繰り返して指定された深さの木を作ったら…

→ 葉にラベルをつける

葉に属すデータ点のうち出現確率が最大のもののラベルを選択（単純ベイズ、多数決）



50

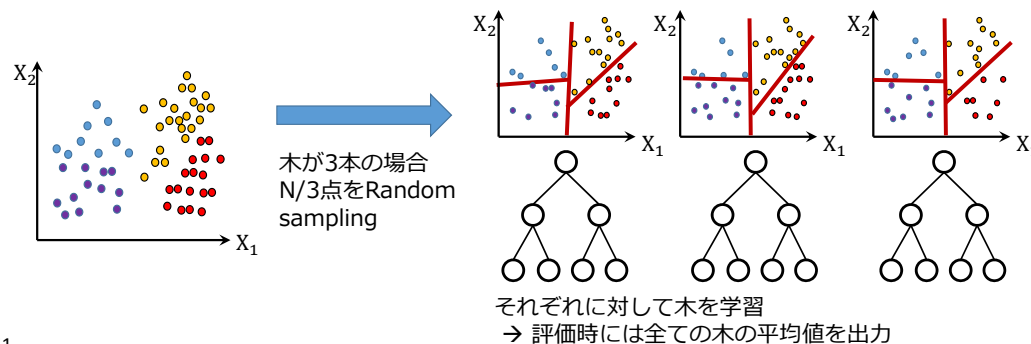
## 集団学習 (Ensemble learning)

弱識別器を多数組み合わせることで強識別器を実現する

弱識別器：精度の低い識別器

強識別器：精度の高い識別器

決定木 → ランダム森(Random Forests)



51

## Support Vector Machine

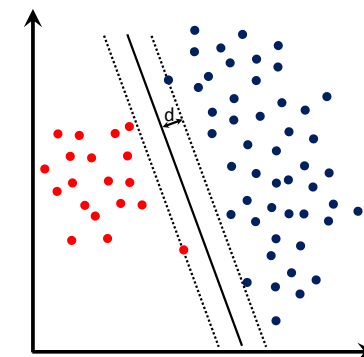
- 特徴空間が超平面（2次元なら直線）で分離可能なとき・・・
- 超平面と最も近いデータ点との距離が最大となるような超平面を選択する
  - これをマージン最大化という
  - 最近傍点をサポートベクトルという
- 超平面の方程式だけを記録すればよいのでメモリ消費が少ない

※線形分離不可能な場合

→ ソフトマージンSVM

→ カーネルトリック

詳細はパターン認識の講義へ



52

# 識別器

- 識別器：教師データに基づき特徴空間を分割することで、未知データへのラベル付けを行なう
- 特に有名な下の識別器を紹介
  - K Nearest Neighbor
  - Random Forests
  - Support Vector Machine

