テキストデータの統計科学入門

第14章　テキストの分類

國井慎也

概要

本章では、テキストをあらかじめ複数グループ（カテゴリ、あるいはクラス）に分けておき、どのグループに属するか不明であるテキストを、いずれかのグループに振り分けるテキストの分類法について紹介する。

古典的な方法

- 線形判別分析
  - 学習データ: $(\boldsymbol{x_1}, y_1), (\boldsymbol{x_2}, y_2), \ldots, (\boldsymbol{x_n}, y_n)$

    $\boldsymbol{x_k} = (x_{k1}, x_{k2}, \ldots, x_{kp})$:テキストから集計した項目

    y_k :テキストが属するカテゴリのラベル

    $Y = \sum_{i=1}^{p} w_i x_i + w_0$ のwを求める
  - 分類方法

    カテゴリが2つの場合:テストデータXを上記の式に代入し、Y>0ならA、Y<0ならBを割り当てる

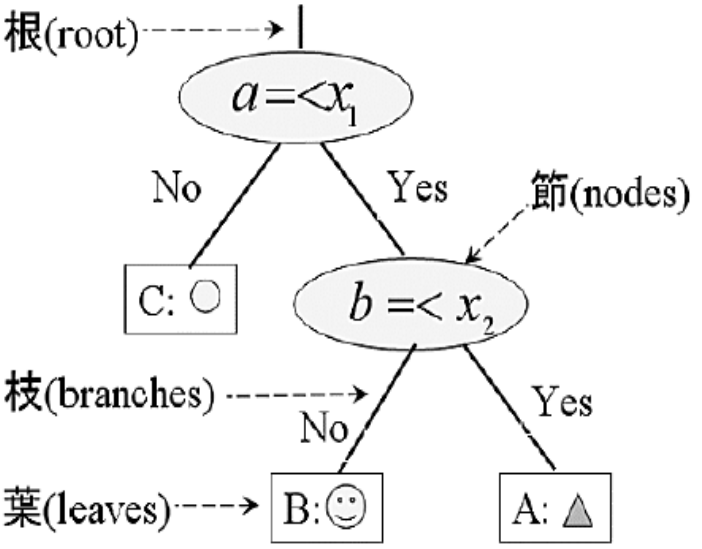
    しかし、欠点として、変数の数やカテゴリの数が多くなると精度はよくない

    距離による判別分析、ベイズ法による判別分析、SVM、K-NN法、ニューラルネットワークなどが提案されている。本書では、決定木による学習について紹介する。

決定木

- 決定木
  - 学習データを用いて変数を分岐させる方法によって分類のルールを構築する手法
  - 線形判別で分割困難な問題でも適用が可能
  - 結果の解釈が容易
  - いくつかアルゴリズムがある
    - CAHID
    - C4.5/C5.0/See5
    - CART

      本書ではこれについて説明



決定木の例

CARTアルゴリズム（１）

- CARTアルゴリズム
  - ２進木を作成する
  - 分岐の際に不純度という指標を使用する
  - 集団学習にも広く使用されている

- 不純度
  - 変数を分岐する前と分岐させた後の誤差の改善を示す指標

$$\Delta GI(t) = P_t GI(t) - P_L GI(t_L) - P_R GI(t_R)$$

P_t, P_L, P_R：それぞれ分割する前、分割した後の左側、分割した後の右側の個体の比率

CARTアルゴリズム（2）

- Gini分散指標
  - 上式のGI(t)

$$GI(t) = 1 - \sum_k p(k \mid t)^2$$

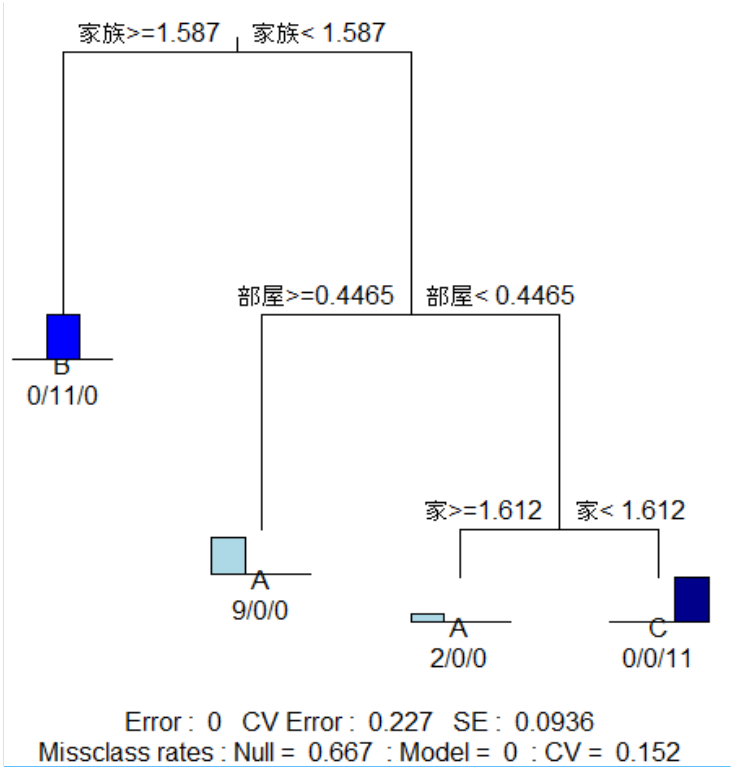
p(k|t):ノードt内のカテゴリkが正しく分類されている比率

上手く分類できれば、Gini分散指標は小さくなり、不純度は大きくなる。

CARTでの決定木は、不純度が大きくなるように変数を選択する。

決定木の作成（1）

- 決定木作成の例
  - 11人が3つのテーマ（A：住まい、B:家族、C：友達）について書いた作文をテーマごとに分類する
  - 作文の中から名詞を取り出し、使用頻度が高い46種類と、それ以外「Others」をまとめた合計47個の変数を使用する



決定木の作成（2）

- Rのコマンド

```
install.packages("mvpart");library(mvpart);
sb3<-read.csv("http://mjin.doshisha.ac.jp/iwanami/data/sb3.csv",head=T,row.names=1)
(sb3.rp<-mvpart(y~.,sb3,size=4))　#sizeは木のターミナルの数
```

- 実行結果

```
n= 33
node), split, n, loss, yval, (yprob)
      * denotes terminal node
 1) root 33 22 A (0.3333333 0.3333333 0.3333333)
   2) 家族>=1.5875 11  0 B (0.0000000 1.0000000 0.0000000) *
   3) 家族< 1.5875 22 11 A (0.5000000 0.0000000 0.5000000)
     6) 部屋>=0.4465 9  0 A (1.0000000 0.0000000 0.0000000) *
     7) 部屋< 0.4465 13  2 C (0.1538462 0.0000000 0.8461538)
      14) 家>=1.612 2  0 A (1.0000000 0.0000000 0.0000000) *
      15) 家< 1.612 11  0 C (0.0000000 0.0000000 1.0000000) *
```

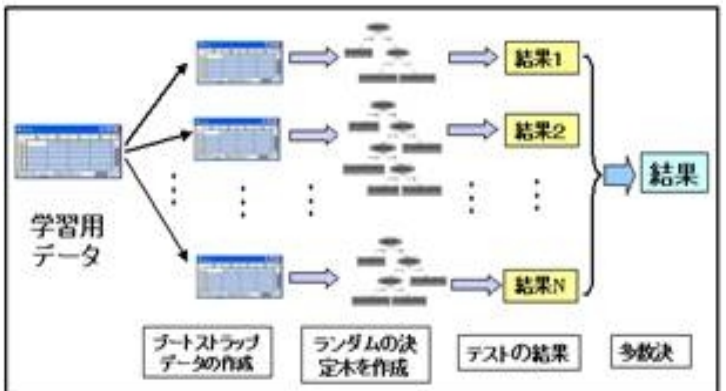
集団学習

- 集団学習（アンサンブル学習）
  - 決して精度が高くない複数の結果を統合・組み合わせて、精度を向上させる方法である
  - 代表的な方法
    - バギング
    - ブースティング
    - ランダムフォレスト
      - 本書では、ランダムフォレストについて紹介されている。
      - 上記の2つの方法よりも計算が速い
      - 上記の2つの方法の分類の精度よりも高いか同等である
      - 大規模のデータマイニングに適している

ランダムフォレスト

- ランダムフォレストアルゴリズム
  - 1. 用いるデータセットから、ブーストラップサンプル$B_1,B_2,\ldots,B_n$を作成する。ただし、構築したモデルを評価するために、1/3のデータを取り除いてサンプリングする。
  - 2. B_kにおけるM個の変数の中からm個ランダムに選ぶ

    $m=\sqrt{M}$　が多用される
  - 3. B_kとm個の変数を用いて未剪定の最大の決定木T_kを作成する
  - 4. それぞれのB_kの決定木T_kにおいて、テストデータでテストを行い、その結果を統合する。
    - 回帰の問題では平均、分類の問題では、多数決をとる



ランダムフォレストの例（１）

11人が3つのテーマ（A：住まい、B:家族、C：友達）について書いた作文をテーマごとに分類する

```
install.packages("randomForest");library(randomForest)
set.seed(10)
(sb3.rf<-randomForest(y~.,sb3))     ＃繰り返し回数は500回
```

```
Call:
 randomForest(formula = y ~ ., data = sb3)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 6

        OOB estimate of  error rate: 9.09%
Confusion matrix:
   A  B  C class.error
A 10  1  0  0.09090909
B  1 10  0  0.09090909
C  1  0 10  0.09090909
```

エラー率

ランダムフォレストの例（２）

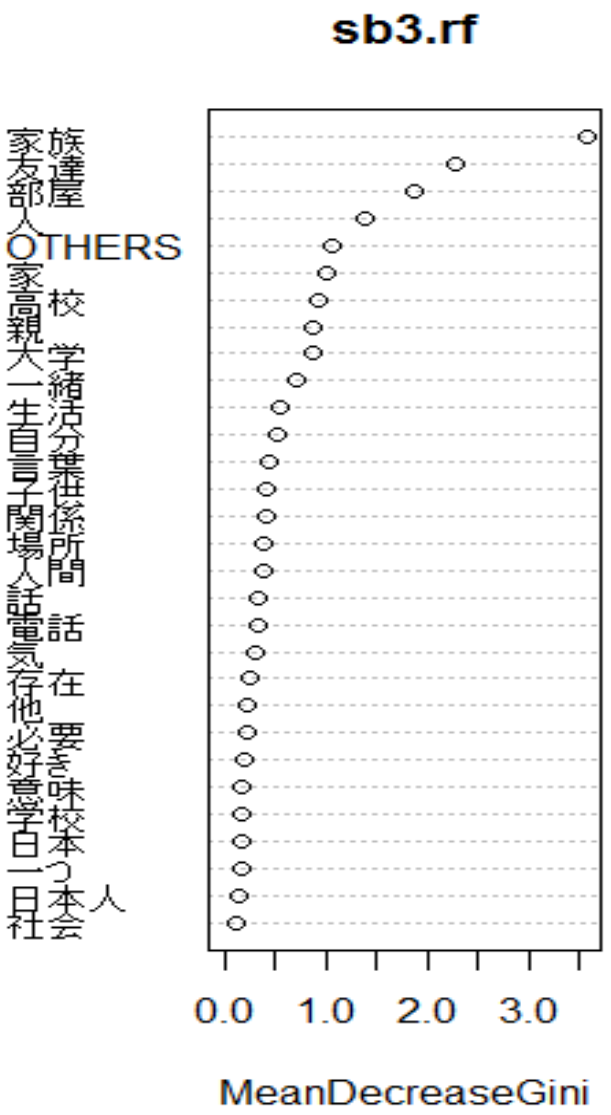
繰り返す回数を50000回にして実行する

```
Call:
 randomForest(formula = y ~ ., data = sb3, ntree = 50000)
               Type of random forest: classification
                     Number of trees: 50000
No. of variables tried at each split: 6

        OOB estimate of  error rate: 6.06%
Confusion matrix:
   A  B  C class.error
A 10  1  0  0.09090909
B  1 10  0  0.09090909
C  0  0 11  0.00000000
```

エラー率が下がっている

変数の重要度



- 決定木を作成される際に計算されたGini係数の平均値
- 変数の重要度を表している

結果の評価指標（1）

カテゴリC_iの分類結果

カテゴリ C_i		分類器の結果	
		Yes	No
データ	Yes	a_i	c_i
	No	b_i	d_i

- 正解率

$$\frac{a_i+d_i}{a_i+b_i+c_i+d_i}$$

- 再現率

分類器がどれくらい「漏れ」なく正しく分別しているかに関する指標

$$R_i=\frac{a_i}{a_i+c_i}$$

- 適合率

分類器の分類結果に混入された「ゴミ」がどれだけ少ないかを表す

$$P_i=\frac{a_i}{a_i+b_i}$$

結果の評価指標（2）

- マクロ平均
  - m個の複数のカテゴリでの分類問題

$$\bar{R}_{ma} = \frac{1}{m}\sum_{i=1}^{m}\frac{a_i}{a_i + c_i}、\quad \bar{P}_{ma} = \frac{1}{m}\sum_{i=1}^{m}\frac{a_i}{a_i + b_i}$$

- マイクロ平均
  - m個の複数のカテゴリでの分類問題

$$\bar{R}_{mi} = \frac{\sum_{i=1}^{m} a_i}{\sum_{i=1}^{m}(a_i + c_i)}、\quad \bar{P}_{mi} = \frac{\sum_{i=1}^{m} a_i}{\sum_{i=1}^{m}(a_i + b_i)}$$

- F_β値
  - 再現率と適合率を折衷した評価指標
  - β＝1であるF_1がよく多用される

$$F_\beta = \frac{(\beta^2 + 1) \times P \times R}{\beta^2 \times P + R}、\quad F_1 = \frac{2 \times P \times R}{P + R}$$

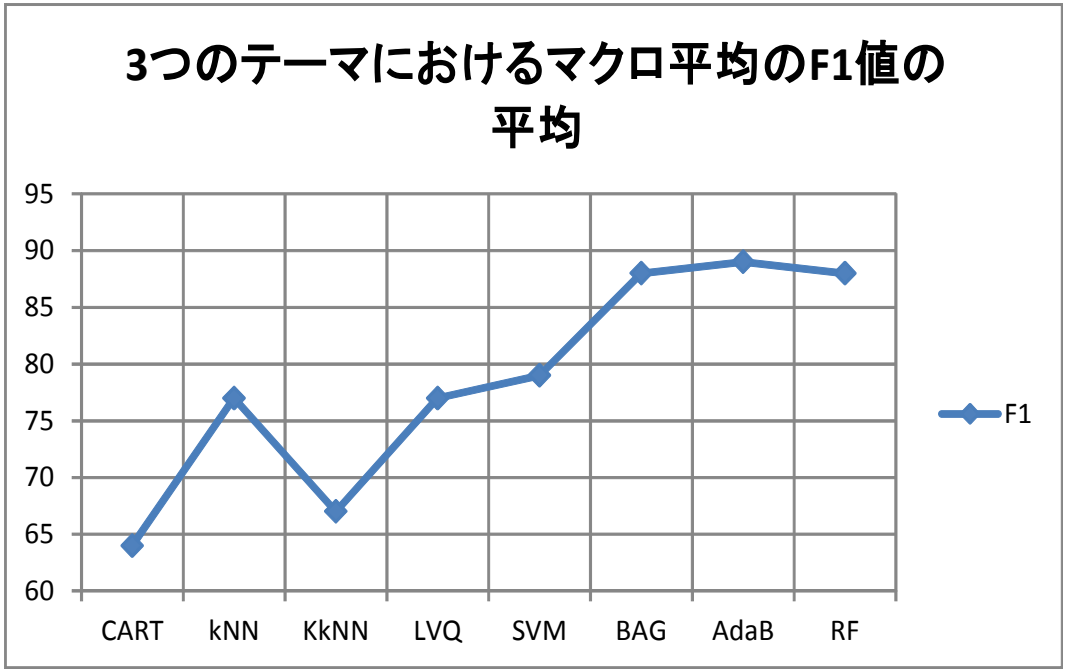
分類器の比較

- はじめに
  - 分類問題において、いくつかの分類器の精度を比較する
  - 分類器の精度は、用いたテキストに依存するので精度の関する評価は絶対的ではないが、多くの研究では、決定木、k-NN法、SVM、ニューラルネット法、ブースティングのパフォーマンスがよいとされている。

- 比較する分類器
  - 決定木（CART）
  - k近隣法（kNN)
  - カーネル法によるk近隣法（KkNN）
  - 量子ベクトル化（LVQ）
  - カーネル法によるサポートベクトルマシン（SVM)
  - バギング（BAG)
  - ブースティング（AdaB)
  - ランダムフォレスト（RF)

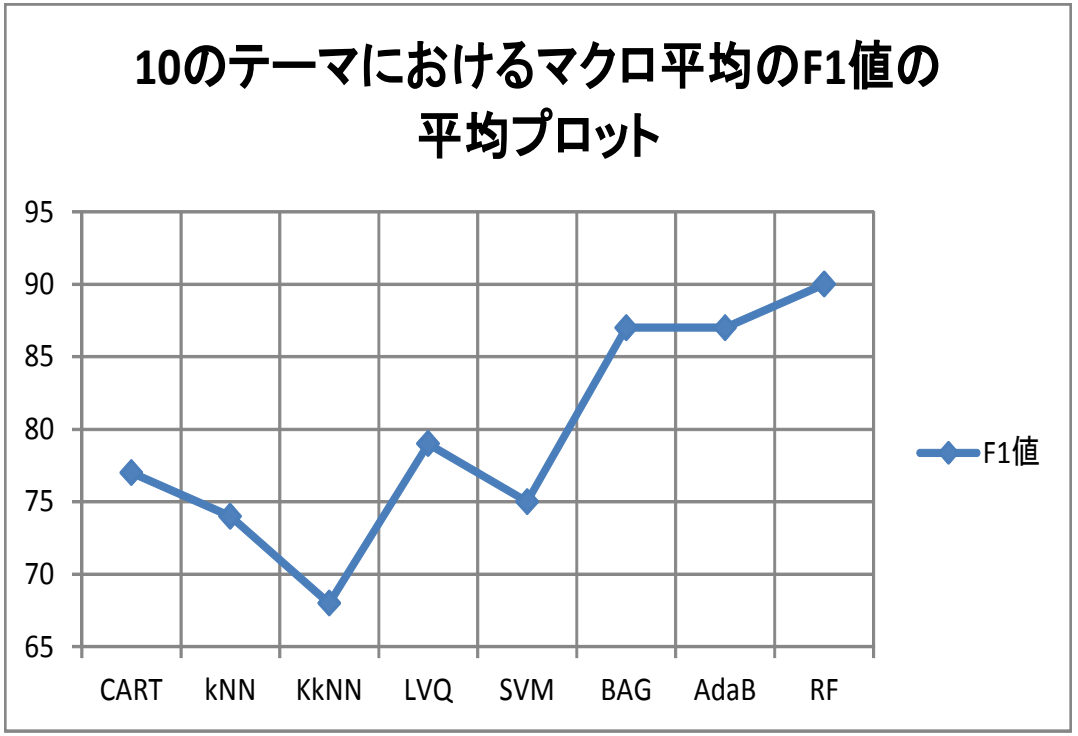
作文の分類（1）

- 3つのテーマの作文に出現する名詞を用いて作文をテーマ別に分類する
  - 作文は、11×3＝33編
  - テストデータは、33編から4編をランダムサンプリングしたもの
  - 学習データは、その残りの29編
  - 学習とテストは、100回繰り返す



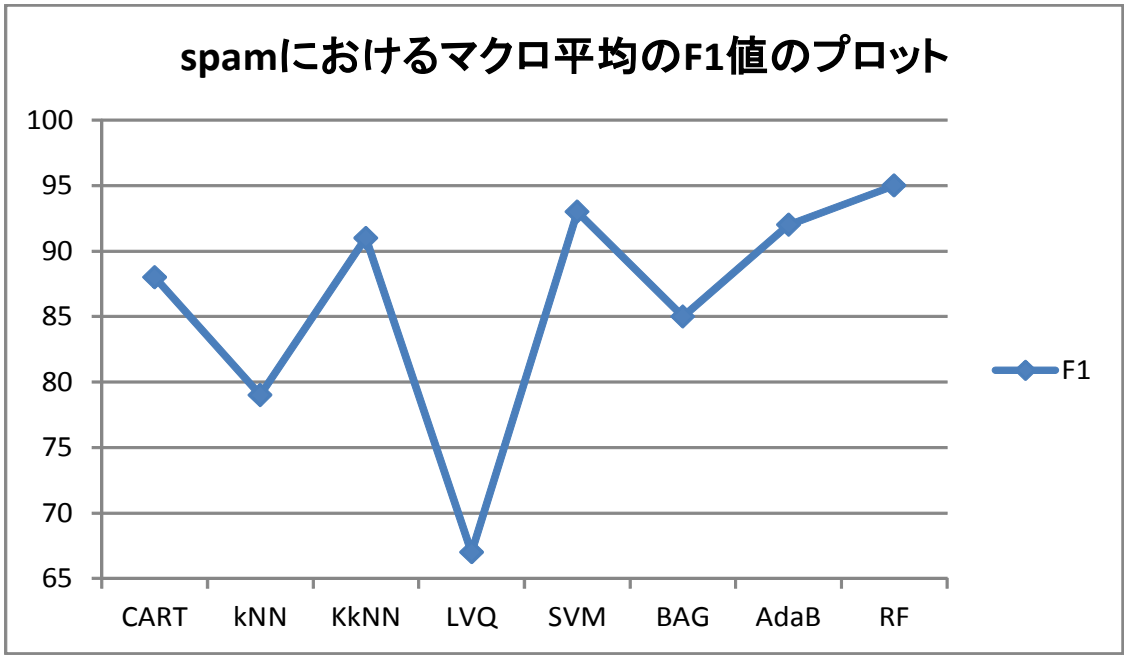
作文の分類(2)

- 10つのテーマの作文に出現する名詞を用いて作文をテーマ別に分類する
  - 作文は、10×11＝110編



スパムメールの判定

- スパムメールの判定
  - Rパッケージ「kernlab」に4601通の英文メールがある
  - スパムメールが1813通、ノンスパムメールが2788通
  - データは57の特徴項目について集計したもの
  - テストデータは、460通をランダムサンプリング
  - 学習とテストは、100回行う



著者の識別

- 文章の著者の識別
  - １０人の書き手の作品それぞれ20編（合計200編）
  - 文章から助詞と読点を集計し、文章の長さに比例する相対頻度に直して用いる
  - テストは、200編の作品を20編をラムダムサンプリング
  - 学習とテストは100回行う