第16章　集団学習

1236640 c

若島　和

集団学習とは・・・

- 精度が高くない複数の結果を組み合わせることで、精度を向上させる方法である

- Ex.) 数学が得意なA・国語が得意なB・英語が得意なC
  3人が力を合わせればテストの正答率があがる

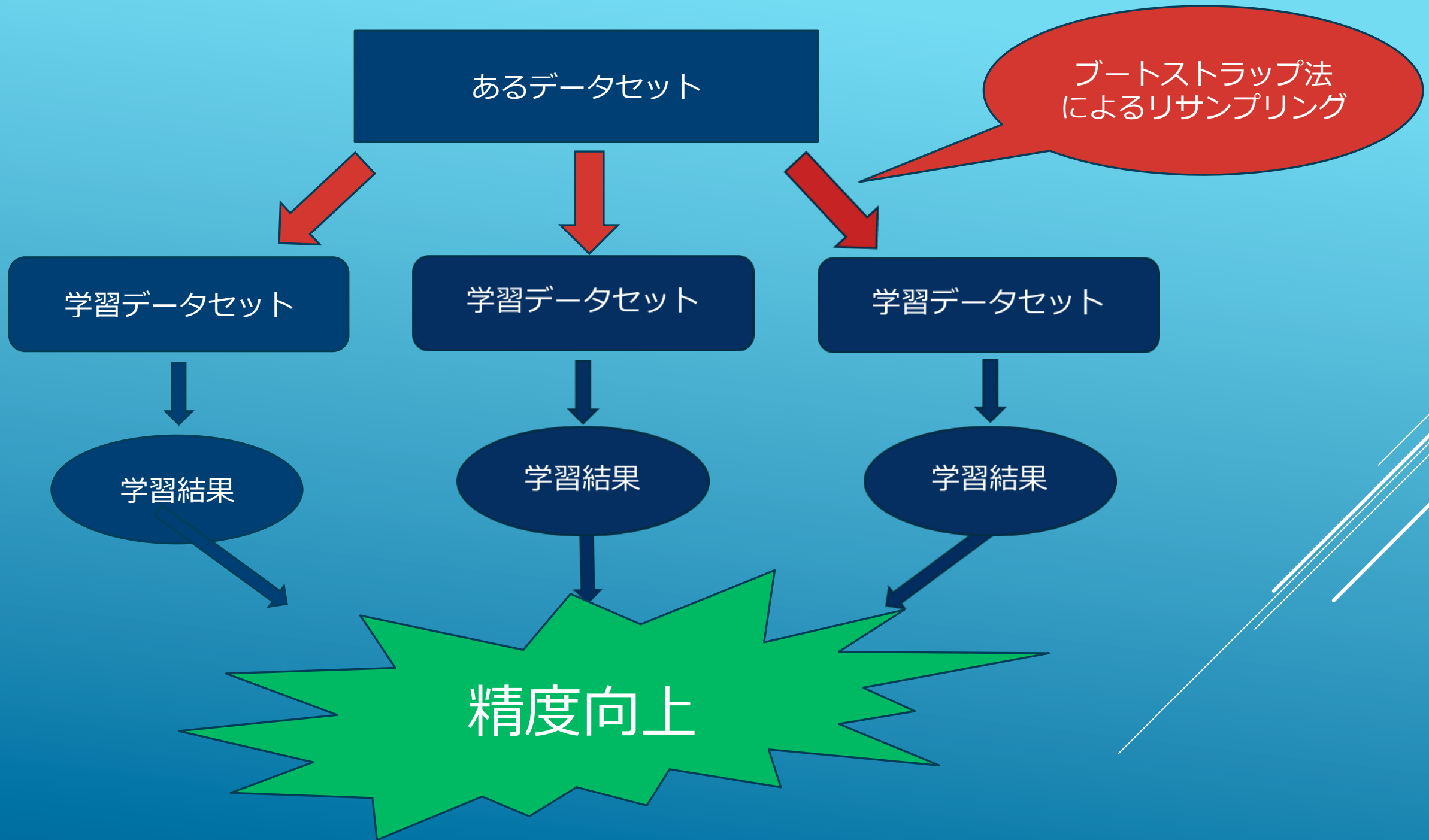
- ①　バギング
- ②　ブースティング
- ③　ランダムフォレスト

集団学習には大きく3種類ある

あるデータから、ブートストラップ法（Bootstrap） というリサンプリング法で複数の学習データを作成

その複数の学習データセットから得た回帰・分類結果を統合・組み合わせることで精度を向上させる

①バギング (BOOTSTRAP AGGREGATING)


あるデータセット
ブートストラップ法
によるリサンプリング
学習データセット
学習データセット
学習データセット
学習結果
学習結果
学習結果
精度向上

Packages(“adabag”)を使う

＊ＣＲＡＮミラーサイトから要ダウンロード

このパッケージにはバギングのアルゴリズムに基づく

関数 bagging がはいっている

パッケージを用いたケーススタディ

```
> library(mlbench)
> data(BrastCancer)
> even.n<-2*(1:349)

> BC.train<-BreastCancer[even.n,-1]  #学習用データ
> BC.test<-BreastCancer[-even.n,-1]  #テスト用データ
> for(i in 1:9){
+ BC.train[,i]<-as.integer(BC.train[,i])
+ BC.test[,i]<-as.integer(BC.test[,i])
+ }
```

パッケージmlbench内にある
データBreastCancerを読み込む

学習用データ・テスト用データに
偶数行・奇数行のデータかで分ける

＊データBreastcancerは、699の乳癌患者について、
11項目に分けて記録したデータフレームである

```
> install.packages("adabag");library("adabag")
> set.seed(20)
> BC.ba<-bagging(Class~ .,data=BC.train)
> BC.bap<-predict(BC.ba,BC.test)
> (tb.ba<-table(BC.test[,10],BC.bap$class))

                benign    malignant
  benign         220          7
  malignant       4          119
> 1-sum(diag(tb.ba))/sum(tb.ba)
[1] 0.03142857     #誤判別率
> sum(diag(tb.ba))/sum(tb.ba)
[1] 0.9685714      #正解率
```

作成した学習用データを用いて
関数baggingで学習を繰り返し、
複数の分析結果を作成する

その複数の結果から
関数predictを用いてテスト用データ
について判別をおこなう

その判別の正解率・誤判別率を計算

与えられたデータをあるモデルで学習し、その学習結果をふまえてその都度データの重みを調整していくことで複数の学習結果を求め、その結果を統合・組み合わせることで精度を向上させる

ブースティングの中で一番有名なのは AdaBoost

というアルゴリズム

AdaBoostにも様々な方法があるが、アルゴリズム共通点は

1．重みの初期値設定
2．重みを用いて学習する
3．誤り率を計算・結果の信頼度を計算
4．それに基づき重みを更新する
5．２～４を繰り返す
6．重みつき多数決で結果を出力する

②ブースティング

先ほどと同じくパッケージ (“adabag”) を使い、乳癌のデータを用いる

このパッケージにはFreud　と Schapier が提案したAdaboost のアルゴリズムに基づく関数 boosting がはいっている

パッケージを用いたケーススタディ

```
> set.seed(20)
> BC.ad<-boosting(Class~.,data=BC.train)
> BC.adp<-predict(BC.ad,newdata=BC.test)
> (res<-BC.adp$confusion)
                Observed Class
Predicted Class benign malignant
      benign       218         3
      malignant      9       120
> 1-sum(diag(res))/sum(res)
[1] 0.03428571      #誤判別率
> sum(diag(res))/sum(res)
[1] 0.9657143     #正解率
```

先のスライドと同じく、乳癌のデータを学習用・テスト用に分けたデータを使い、

関数predictを用いてテスト用データを判別し、その正解率・誤判別率を計算

20世紀になり提案された最新のデータ解析方法である
精度・計算機資源の節約などの面で
バギング・ブースティングより優れている

『アルゴリズム』

１．与えられたデータから複数のブートストラップサンプルを作成

２．各々のブートストラップサンプルデータを用いて最大の決定・回帰木を生成する（分岐のノードはランダムサンプリングされた変数の中の最善のもの）

３．すべての結果を統合・組み合わせて新しい予測・分類器を作成する

③ランダムフォレスト（RF）

バギングではすべての説明変数を用いるのに対し、
ランダムフォレストでは説明変数をランダムサンプリングして用いる

これにより、ランダムフォレストは高次元データ解析に向いている

ランダムフォレストとバギングの違い

ランダムフォレスト専用のパッケージrandomForestを使う
このパッケージにはランダムフォレスト分析ができる
関数randomForestがはいっている

関数

randomForest (formula, data = NULL, ..., ..., subset, na.action=na.fail)

主要な引数	内容
formula	モデルの形式
x, y	目的変数と説明変数 (formula 代わりに用いる)
data, subset	用いるデータ
na.action	欠損値の表記型の指定
ntree	生成する木の数 (デフォルトは 500)
mtry	分岐に用いる変数の数 (デフォルト, 分類 √M, 回帰 M/3, M:変数総数)
importance	変数の重要度出力 (デフォルトは FALSE)

パッケージを用いたケーススタディ

```
> install.packages("randomForest");library("randomForest")
> set.seed(20)
> BC.rf<-randomForest(Class~.,data=BC.train,na.action="na.omit")
> print(BC.rf)

Call:

 randomForest(formula = Class ~ ., data = BC.train, na.action = "na.omit")
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 3

        OOB estimate of  error rate: 3.25%
Confusion matrix:
          benign malignant class.error
benign       214         7  0.03167421
malignant      4       113  0.03418803
```

```
> summary(BC.rf)
                Length Class  Mode
call              4    -none- call
type              1    -none- character
predicted       338    factor numeric
err.rate       1500    -none- numeric
confusion         6    -none- numeric
votes           676    matrix numeric
oob.times       338    -none- numeric
classes           2    -none- character
importance        9    -none- numeric
importanceSD      0    -none- NULL
localImportance   0    -none- NULL
proximity         0    -none- NULL
ntree             1    -none- numeric
mtry              1    -none- numeric
forest           14    -none- list
y               338    factor numeric
test              0    -none- NULL
inbag             0    -none- NULL
terms             3    terms  call
na.action        11    omit   numeric
```

```
> BC.rf$type
[1] "classification"
> plot(BC.rf)
> BC.rf$importance
                  MeanDecreaseGini
Cl.thickness          8.4980928
Cell.size            34.2300017
Cell.shape           26.3917432
Marg.adhesion         4.7446165
Epith.c.size         15.3275428
Bare.nuclei          21.3485556
Bl.cromatin          23.3843991
Normal.nucleoli      17.5526990
Mitoses               0.8232664
 varImpPlot(BC.rf)

> (1-sum(diag(BC.rft))/sum(BC.rft))
[1] 0.02898551
> (sum(diag(BC.rft))/sum(BC.rft))
[1] 0.9710145
```