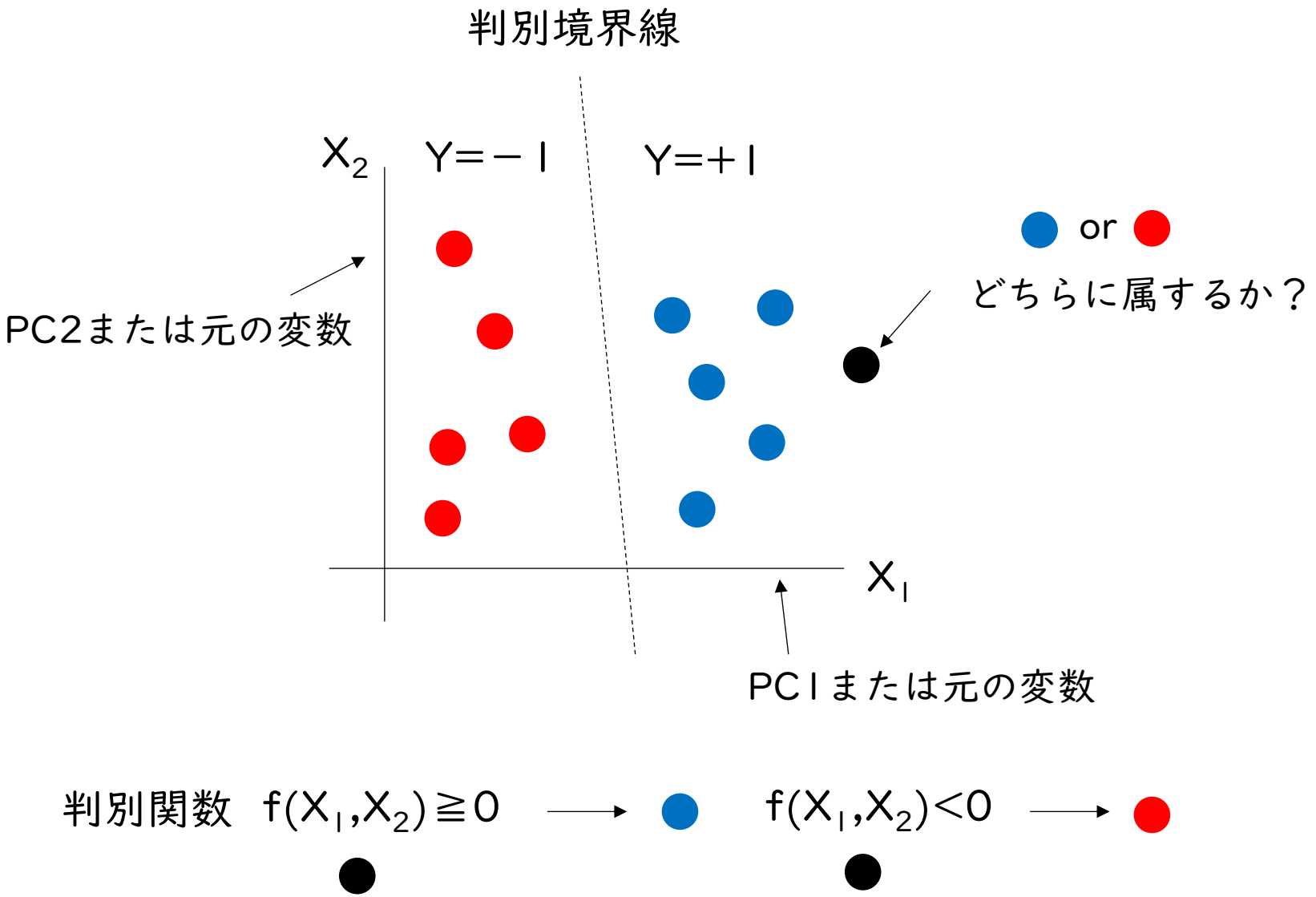
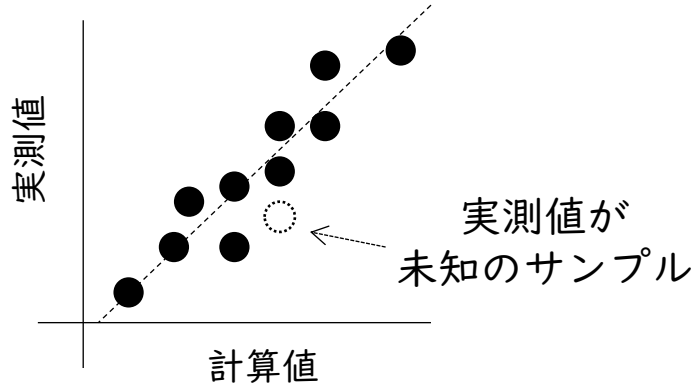
判別分析

(線形)判別分析のイメージ

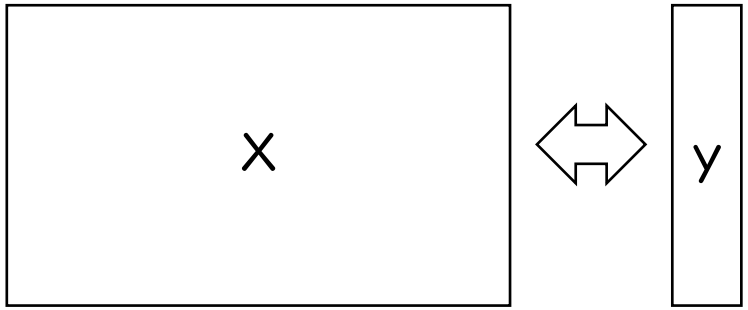


回帰・判別モデルの共通点

- PLS回帰：回帰モデル

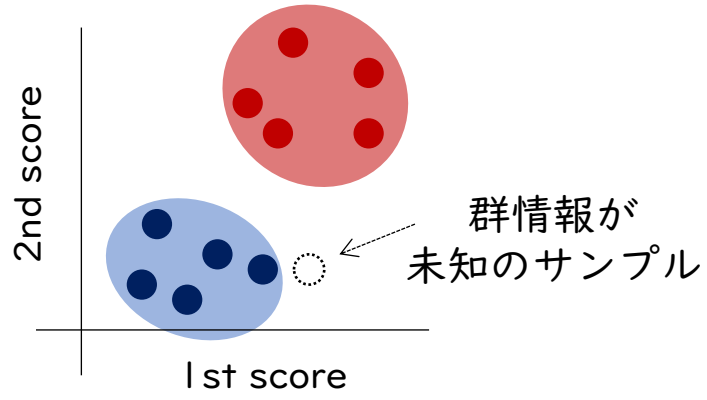


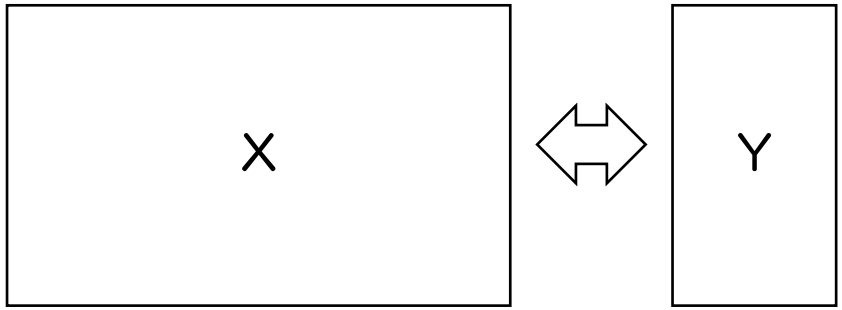
判別モデルも回帰モデルでも、
PLSの理論としてはほとんど同じ



連続値のベクトル

- PLS-DA：判別モデル

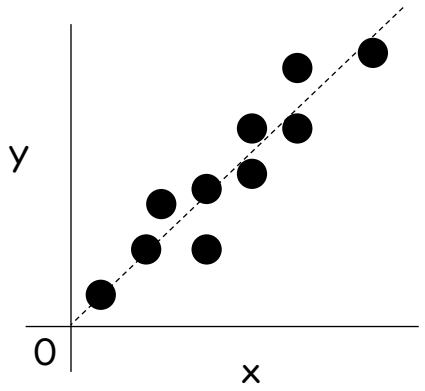




群の情報を表す
0と1の行列

[再掲]回帰分析のモデル

- 単回帰分析



目的変数　　1変数　　誤差

$$y = \sum_{n(\text{サンプル})} \{ x \times b + e \}$$

(原点を通るとき)

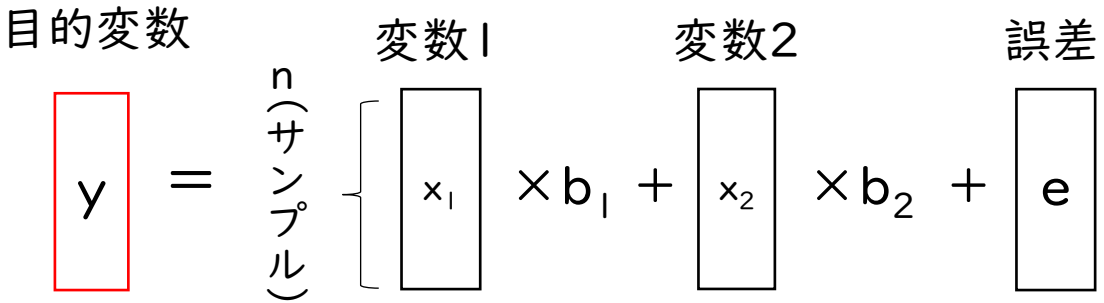
- 重回帰分析(2変数の場合)

目的変数　　変数1　　変数2　　誤差

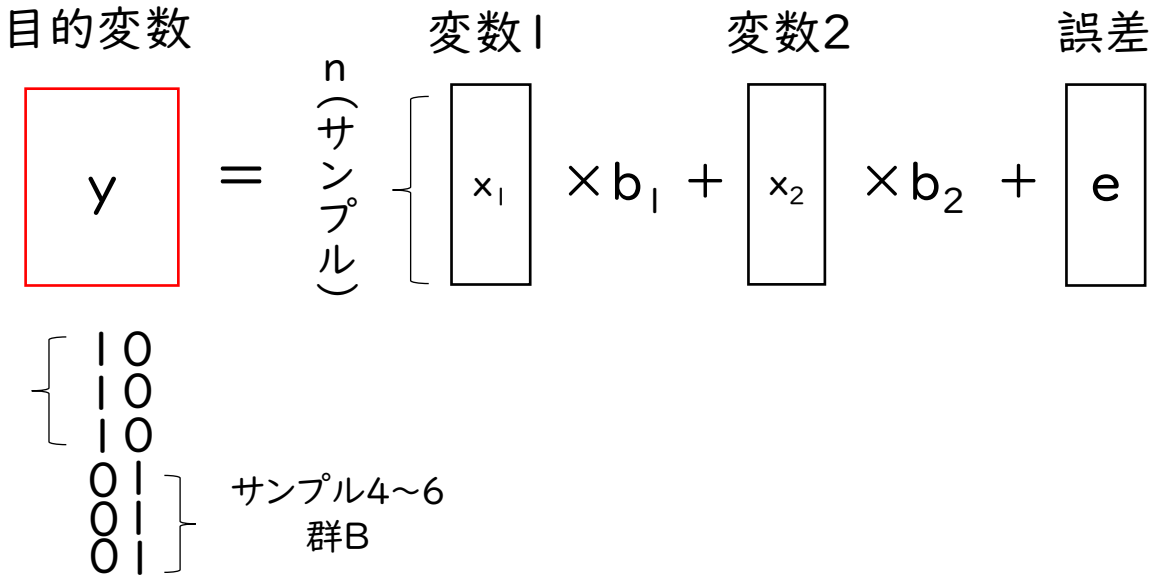
$$y = \sum_{n(\text{サンプル})} \{ x_1 \times b_1 + x_2 \times b_2 + e \}$$

回帰分析と判別分析の違い

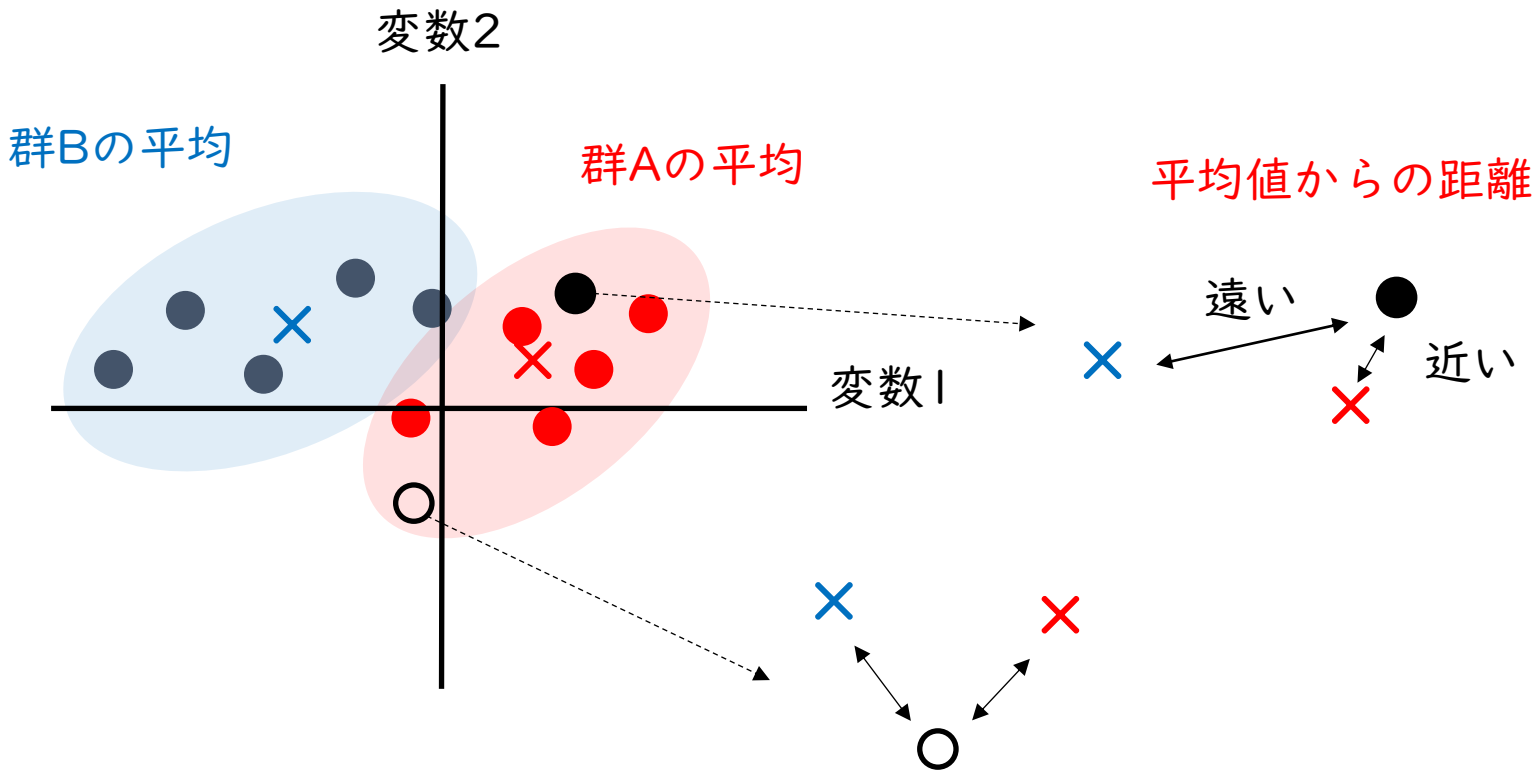
- 回帰分析(2変数の場合)



- 判別分析(2変数の場合)



判別分析のイメージ(2群の場合)

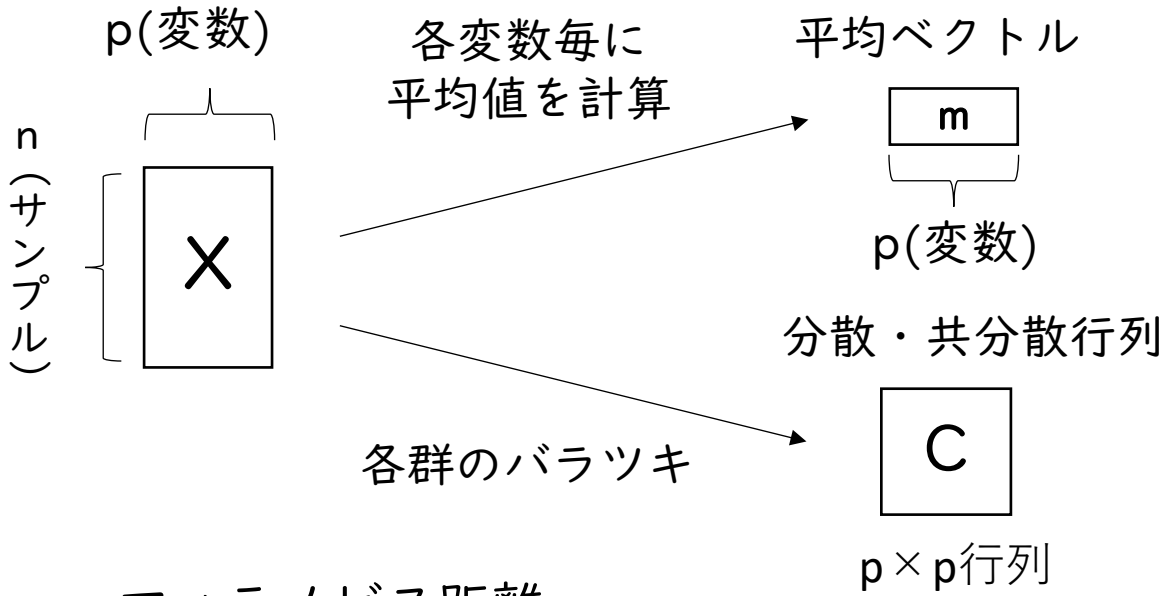


平均値からの距離は同程度だが、
赤に近いように見える

各群のバラツキの様子を見ると、○は●に近いと判断できる

判別分析(2群の場合)

群A、群Bのそれぞれについて
平均ベクトルと分散・共分散行列を計算



マハラノビス距離

未知のサンプル

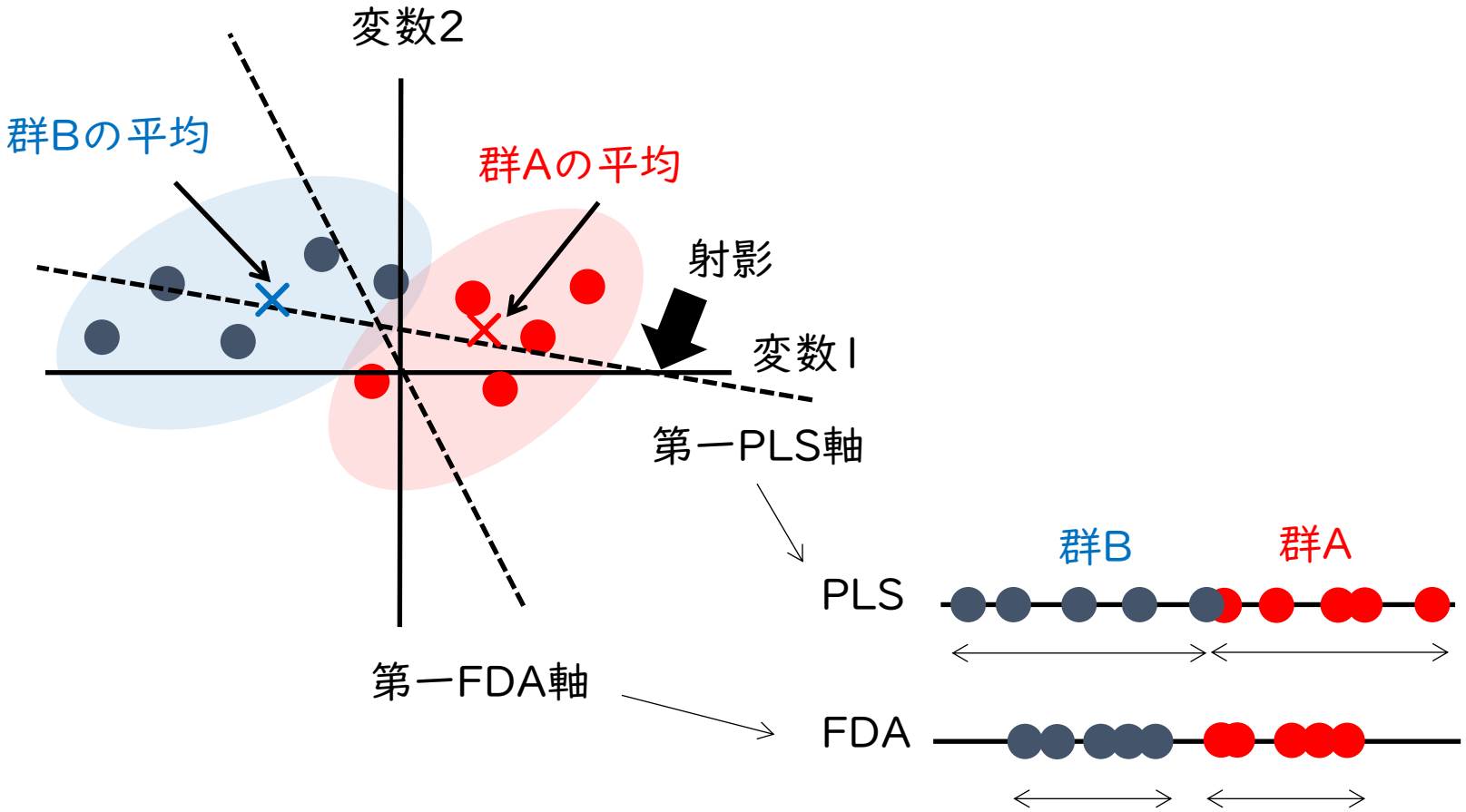
x

群A $\sqrt{(\underline{\mathbf{x}}-\mathbf{m}_1)^t\mathbf{C}_1^{-1}(\underline{\mathbf{x}}-\mathbf{m}_1)}$

群B $\sqrt{(\underline{\mathbf{x}}-\mathbf{m}_2)^t\mathbf{C}_2^{-1}(\underline{\mathbf{x}}-\mathbf{m}_2)}$

距離が近い方の
群に割り当てる

フィッシャーの線形判別分析(FDA)



データに依存するが、
FDAの方が群内のバラツキが小さい

FDAは、スコアの相関係数を最大化する方法

- FDA

$$\mathbf{t} = \mathbf{X}\mathbf{w}_x \quad \mathbf{s} = \mathbf{Y}\mathbf{w}_y$$

$$corr(\mathbf{t},\mathbf{s}) = \frac{\mathrm{cov}(\mathbf{t},\mathbf{s})}{\sqrt{\mathrm{var}(\mathbf{t})}\sqrt{\mathrm{var}(\mathbf{s})}}$$ を最大化

(正準相関分析の定式化)

スコアtとsの共分散が大きい
かつスコアtまたはsの分散が小さい
→ 群内のバラツキが小さいのと同じ

3群 N=3のとき

$$\mathbf{Y} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

- PLS

$\mathrm{cov}(\mathbf{t},\mathbf{s})$ を最大化

スコアtとsの共分散が大きい

FDAとPLSはいずれも群間差が大きく、
FDAはさらに群内のバラツキを小さくする方向が得られる

正則化フィッシャー線形判別分析

- FDA

$$\mathbf{X'Y(Y'Y)^{-1}Y'Xw_x} = \lambda \mathbf{X'Xw_x}$$

$$\mathbf{Y'X(X'X)^{-1}X'Yw_y} = \lambda \mathbf{Y'Yw_y}$$

p>nだと逆行列の計算が出来ない

- 正則化FDA

$$\mathbf{X'Y(Y'Y)^{-1}Y'Xw_x} = \lambda \left\{(1-k)\mathbf{X'X} + k\mathbf{I}\right\} \mathbf{w_x}$$

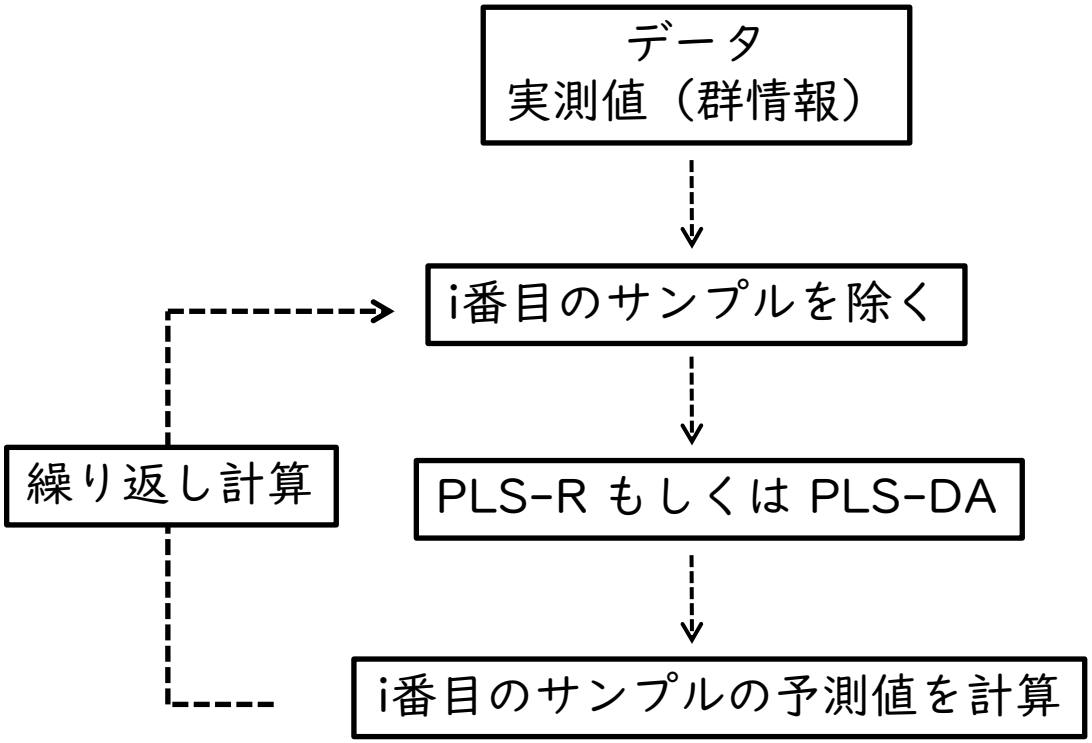
$$\mathbf{Y'X}\left\{(1-k)\mathbf{X'X} + k\mathbf{I}\right\}^{-1} \mathbf{X'Yw_y} = \lambda \mathbf{Y'Yw_y}$$

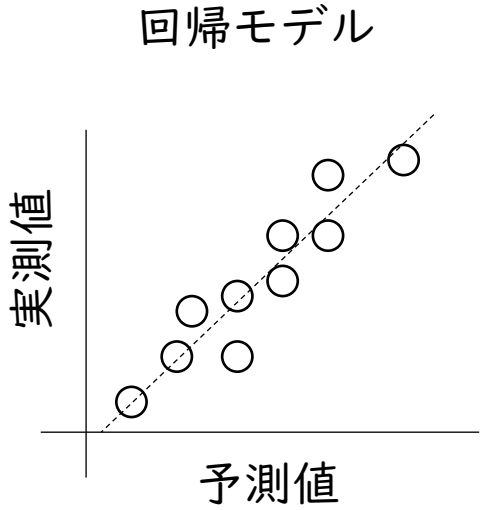
p>nでも計算可能

重回帰分析とリッジ回帰の関係と同じように、
逆行列の計算に単位行列を導入することで計算出来る

[再掲] 判別モデルの性能評価 (回帰と同じ)

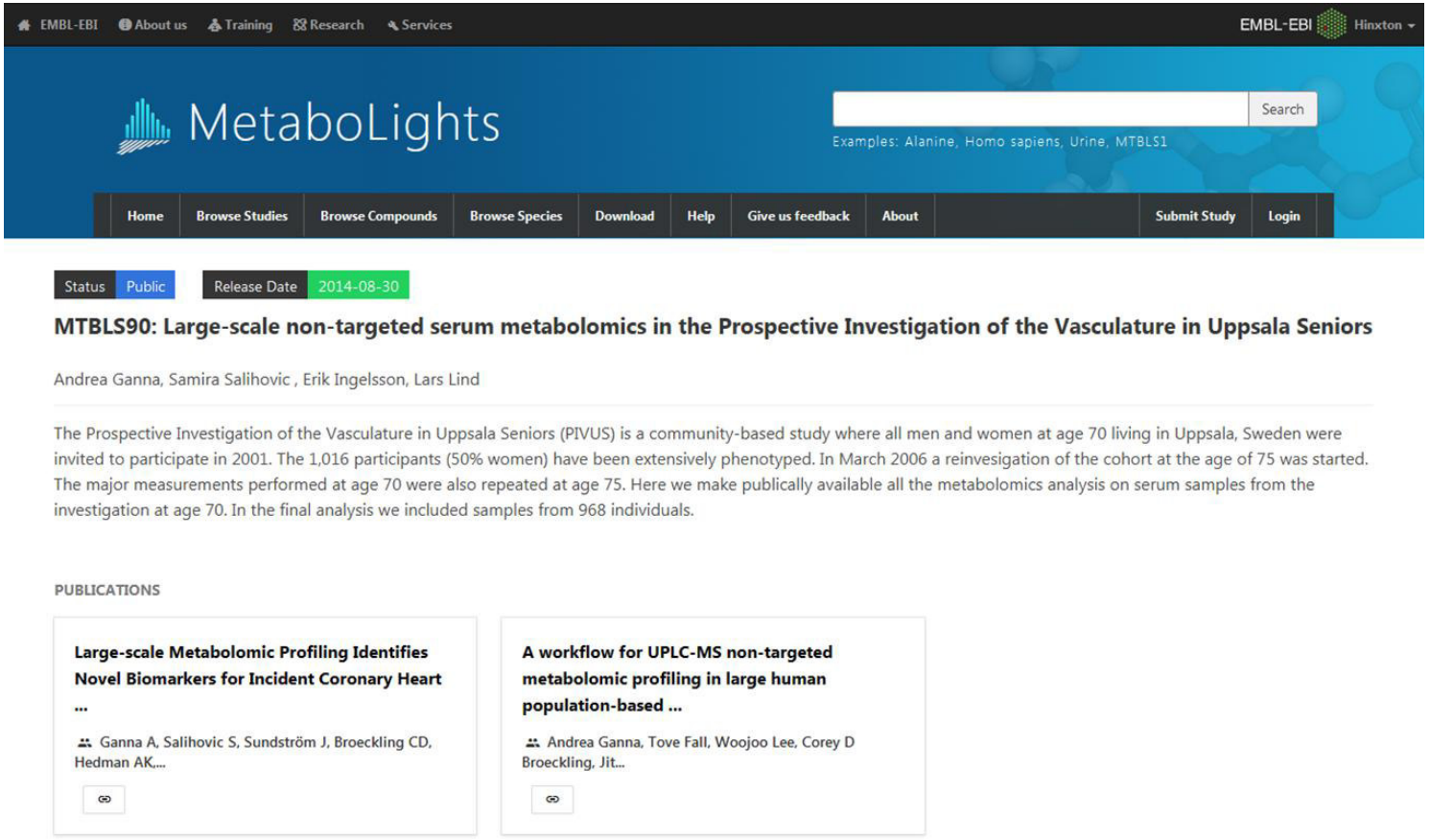
例. Leave-one-out cross validation (LOOCV)の場合





正しく性能評価を行うためには、さらに独立テストが必要

解析に用いるデータ



968サンプル(女性:483名 男性:485名)
189物質のメタボロームデータ

PLS判別分析(1)

- データの読み込み

```
X0 <- read.csv(file="C:/R/MTBLS90_train.csv")
```

- データの準備

```
X <- as.matrix(X0[,-c(1,2)])  # メタボロームデータ
Y0 <- X0[,2] # 群情報(性別)
Y <- model.matrix(~ Y0 + 0) # 目的変数
```

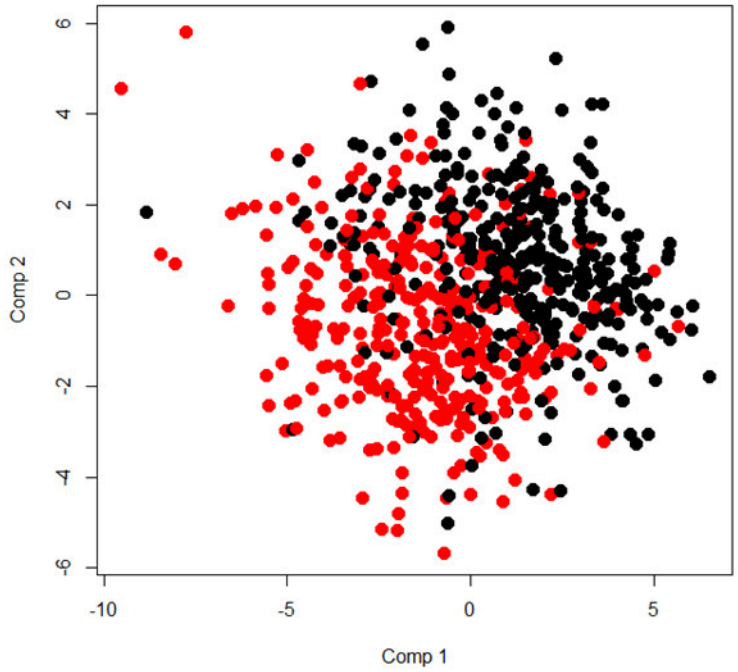
- PLS判別分析

```
library(chemometrics)
plsda <- cppls(Y~X, 30, data=data.frame(X=X, Y=Y),
                                                scale=TRUE)
```

PLS判別分析(2)

- PLS判別分析のスコア

```
col_class <- NULL;
col_class[Y0=="Male"] <- 1; col_class[Y0=="Female"] <- 2
plot(plsda$scores, col=col_class, pch=16, cex=1.5)
```



PLS判別分析の比較

- chemometricsパッケージ

```
library(chemometrics)
plsda1 <- cppls(Y~X, 2, data=data.frame(X=X, Y=Y),
                                                scale=TRUE)
```

- plsパッケージ

```
library(pls)
plsda2 <- pls2_nipals(X, Y, a=2, scale=TRUE)
```

- mixOmicsパッケージ

```
library(mixOmics)
plsda3 <- plsda(X, Y0) # デフォルトでscale=TRUE
```

PLS判別分析(3)

- クロスバリデーション(10 fold)

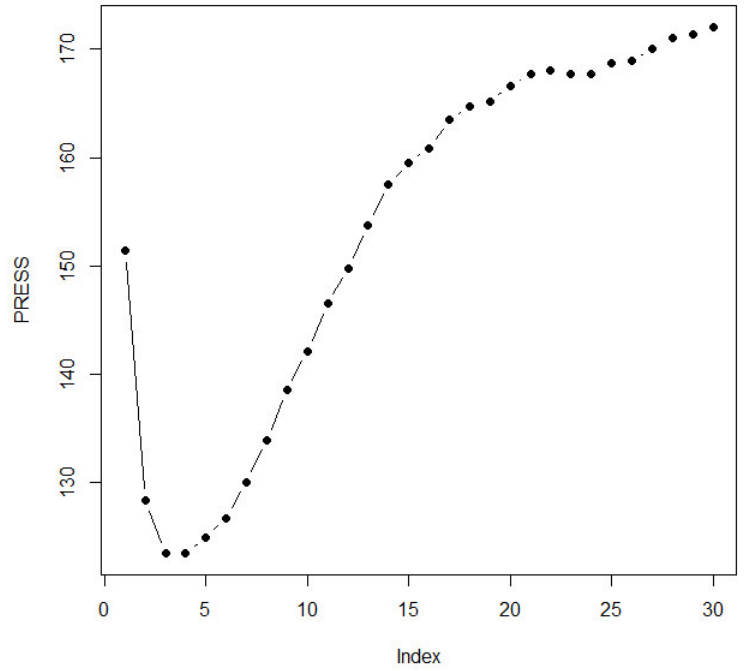
```
plsda <- cppls(Y~X, 30, data=data.frame(X=X, Y=Y),
validation="CV", segment=10, segment.type="consecutive",
scale=TRUE)
```

- 潜在変数の数の決定

```
PRESS <- plsda$validation$PRESS[1,]
plot(PRESS, type="b")
```

PLS判別分析(4)

- 潜在変数の数の決定



lvnum <- which.min(PRESS)　　# 3がPRESS最小

PLS判別分析(5)

- テストデータの読み込み

```
Z0 <- read.csv(file="C:/R/MTBLS90_test.csv")
```

- データの準備

```
Z <- as.matrix(Z0[,-c(1,2)])  # メタボロームデータ
Ytest <- Z0[,2] # テストデータの性別
```

- 予測

```
p0 <- predict(plsda, ncomp=lvnum, newdata=Z)
```

PLS判別分析(6)

- 予測結果の整理

```
p1 <- max.col(p0[,,1])
p <- NULL;
p[which(p1==1)] <- "Female";  p[which(p1==2)] <- "male"
```

- 集計結果

```
table(Ytest, p)
```

		予測結果		
		Female	male	合計
正解	Female	98	22	120
	Male	29	92	121

正解率は女性で81.7%、男性で76.0%

Metabolomics (2019) 15:150
https://doi.org/10.1007/s11306-019-1612-4

ORIGINAL ARTICLE

A comparative evaluation of the generalised predictive ability of eight machine learning algorithms across ten clinical metabolomics data sets for binary classification

Kevin M. Mendez[1] · Stacey N. Reinke[1] · David I. Broadhurst[1]

Table 1 The ten data sets curated for this study

Study ID	Publication	Platform	Type	No. of samples (case/control)	No. of peaks	Case/control
MTBLS90[a]	Ganna et al. (2014); Ganna et al. (2015)	LC–MS	Plasma	968 (485/483)	189	Sex (M/F)
MTBLS92[a]	Hilvo et al. (2014)	LC–MS	Plasma	253 (142/111)	138	Breast cancer chemotherapy (before/after)
MTBLS136[a]	Stevens et al. (2018)	LC–MS	Serum	668 (337/331)	689	Postmenopausal hormone (estrogen/estrogen + progesterone)
MTBLS161[a]	Armstrong et al. (2015)	NMR	Serum	59 (34/25)	29	Chronic fatigue syndrome (case/control)
MTBLS404[a]	Thévenot et al. (2015)	LC–MS	Urine	184 (101/83)	120	Sex (M/F)
MTBLS547[a]	Zheng et al. (2017)	LC–MS	Caecal	97 (46/51)	42	High fat diet (case/control)
ST000369*	Fahrmann et al. (2015)	GC–MS	Serum	80 (49/31)	181	Adenocarcinoma (case/control)
ST000496*	Sakanaka et al. (2017)	GC–MS	Saliva	100 (50/50)	69	Debridement (pre/post)
ST001000*	Franzosa et al. (2019)	LC–MS	Stool	121 (68/53)	747	Inflammatory bowel diseases (Crohn's disease/ulcerative colitis)
ST001047*	Chan et al. (2016)	NMR	Urine	83 (43/40)	149	Gastric cancer (gastric cancer/healthy)

*Indicates data sourced from Metabolomics Workbench (https://www.metabolomicsworkbench.org)

[a]Indicates data sourced from Metabolights (https://www.ebi.ac.uk/metabolights/)

機械学習の計算の流れ

1. 訓練データとテストデータに分割する
2. 訓練データでモデル作成
3. クロスバリデーション
4. テストデータで予測

訓練データとテストデータに分割する

- パッケージの読み込み

```
library(caret)
```

- csvファイルの読み込み

```
X0 <- read.csv(file="C:/R/data/MTBLS90.csv")
```

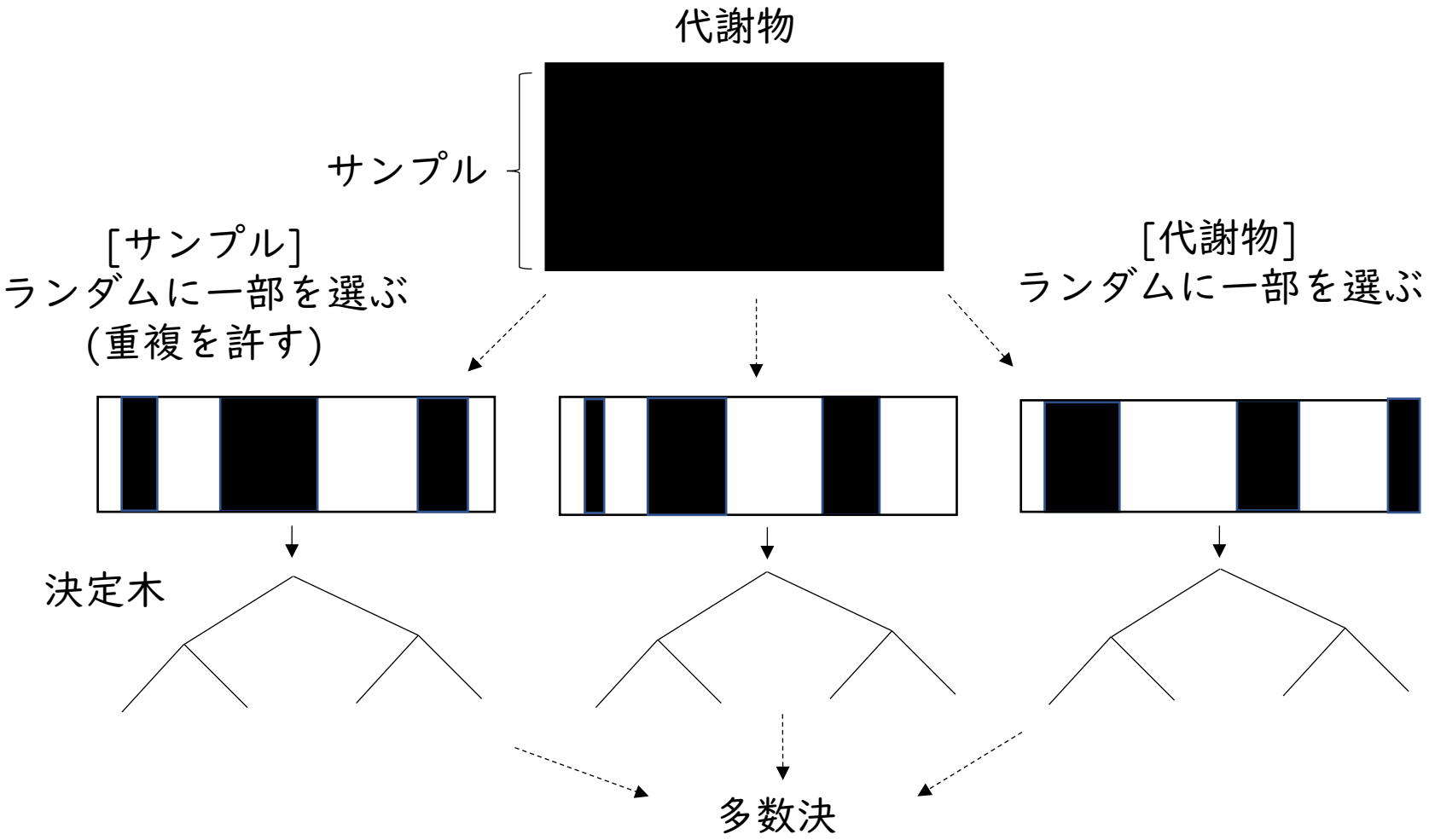
- データの準備

```
X <- as.matrix(X0[,-c(1:4)])
y <- X0[,4]
```

- データの分割(3/4を訓練データ、1/4をテストデータ)

```
trainIndex <- createDataPartition(y, p = .75, list=FALSE)
train_data <- data.frame(X[trainIndex,],
                                    y=factor(y[trainIndex]))
test_data <- data.frame(X[-trainIndex,], y=factor(y[-trainIndex]))
```

ランダムフォレストの概要



- caretパッケージでのランダムフォレストの計算

cv_rf <- train(・・・ (省略), method = “rf", ・・・ (省略)))

ランダムフォレスト

- モデル作成(ランダムフォレスト)

```
set.seed(1)
cv_rf <- train(
    target ~ .,
    data = data.frame(train_data[, -ncol(train_data)],
                                        target=train_data$y),
    method = "rf",
    trControl = trainControl(method = "cv", number=10,
                savePredictions = TRUE, classProbs=T)
)
```

クロスバリデーションで性能評価

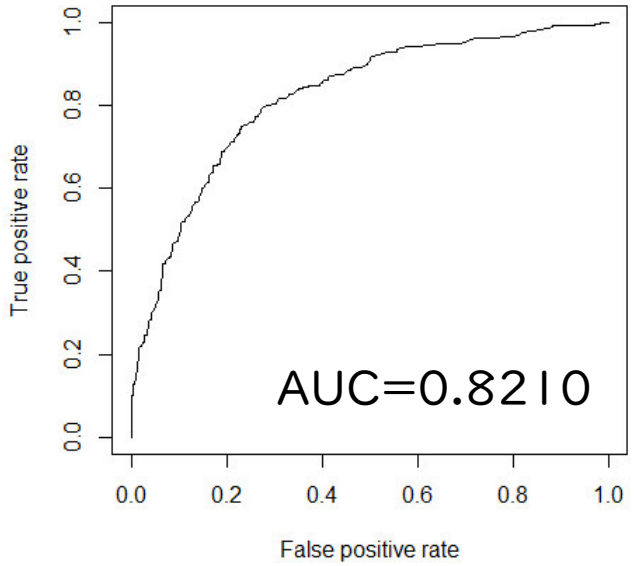
- クロスバリデーション

```
library(ROCR) # ROC curve
pred_cv <- cv_rf$pred
index_cv <- which(pred_cv$mtry == cv_rf$bestTune$mtry)

pred <- prediction(pred_cv[index_cv,]$Male, pred_cv[index_cv,]$obs)
perf_cv <- performance(pred, "tpr", "fpr")
plot(perf_cv)
```

- AUC

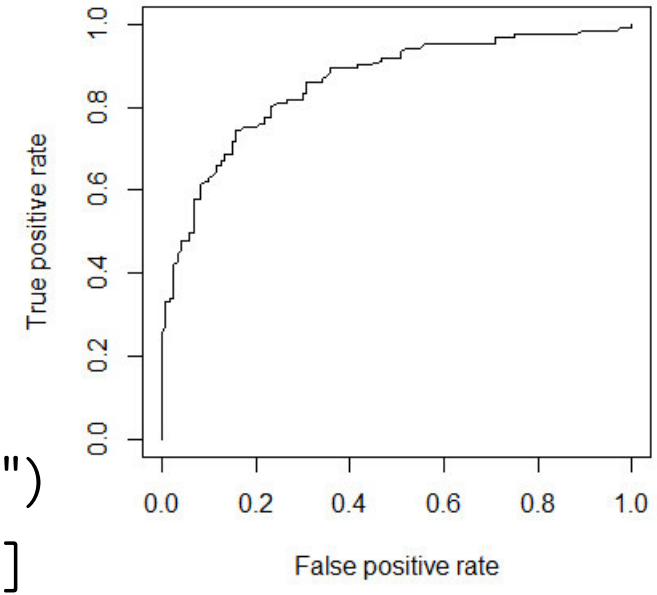
```
auc.tmp <- performance(pred,"auc")
auc_cv <- auc.tmp@y.values[[1]]
```



テストデータに対する予測

- ROC曲線

```
p <- predict(cv_rf, newdata=data.frame(
                    test_data[,-ncol(test_data)]), type="prob")
pred <- prediction(p[,2], test_data$y)
perf_test <- performance(pred, "tpr", "fpr")
plot(perf_test)
```



- AUC

```
auc.tmp <- performance(pred,"auc")
 auc_test <- auc.tmp@y.values[[1]]
```

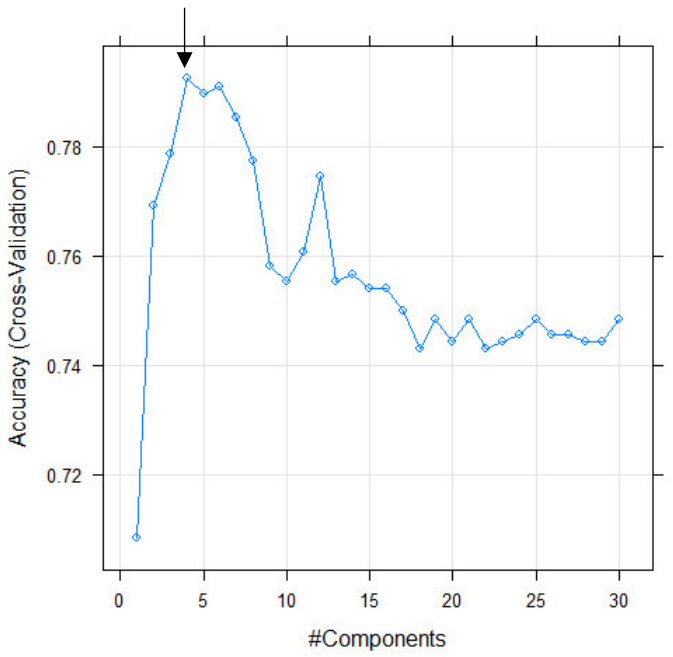
AUC=0.8579

PLS判別分析 caretの場合

- クロスバリデーション

```
cv_plsda <- train(
   target ~ .,
   data = data.frame(
         train_data[, -ncol(train_data)],
         target=train_data$y),
   method = "pls",
   trControl = trainControl(
      method = "cv",
      number=10,
      savePredictions = TRUE,
     classProbs=T),
   tuneLength = 30
)
```

4が最適



plot(cv_plsda)

Boostingアルゴリズムの概要

- 各サンプルについてデータと重みをセットで考える
- 以下の手順を繰り返す
  - 単純な方法(弱学習)で学習を行う
  - 性能評価を行って、重みを変更する
- それぞれに予測を行い、多数決で結果を統合する

- Gradient Boosting(勾配ブースティング)

```
cv_gbm <- train(・・・ (省略), method = “gbm", ・・・ (省略)))
```

Elastic net (Lasso)の概要

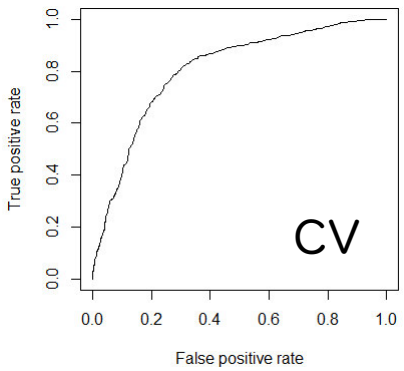
- 変数の数がサンプルの数よりも多い場合、回帰や判別が計算が出来ない(先述のリッジ回帰、FDAなど)
  - リッジ回帰では重みwのL2ノルム正則化 ||w||
  - LassoはwのL1ノルムの正則化|w|
  - Elastic netはwのL1とL2ノルム両方の正則化
    ||w||+|w|

- Elastic net (Lasso)

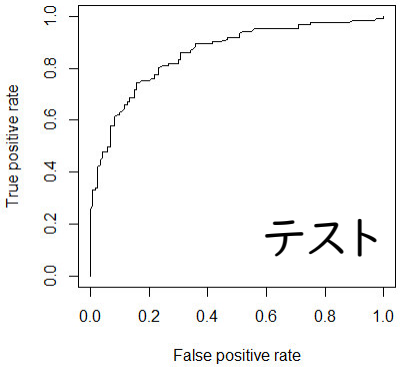
  cv_glmnet <- train(・・・ (省略), method = “glmnet", ・・・ (省略)))

予測結果の比較

ランダムフォレスト

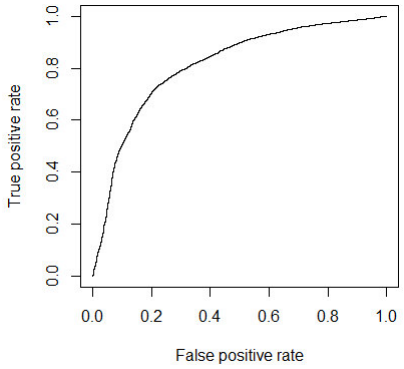


AUC: 0.8210

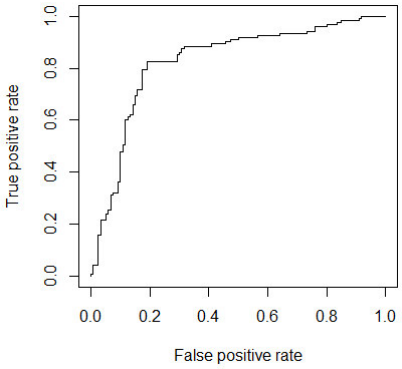


AUC: 0.8579

PLS-DA

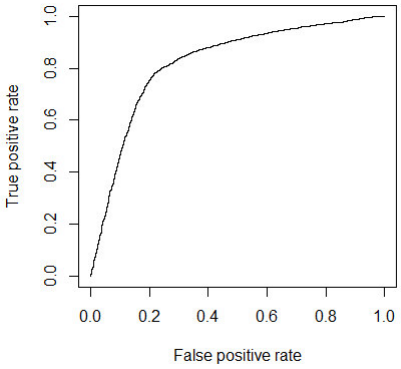


AUC: 0.8111

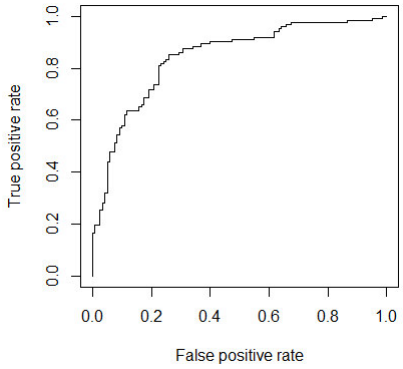


AUC: 0.8256

Gradient Boosting

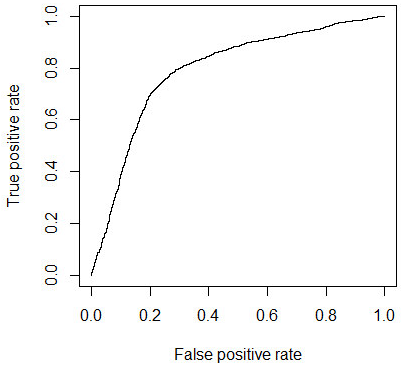


AUC: 0.8222

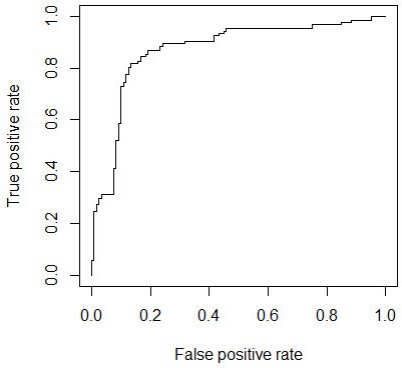


AUC: 0.8426

Elastic net



AUC: 0.7913



AUC: 0.8683

どの方法でも同程度の予測精度が得られており、PLS-DAとも大差はない

各手法での重要な変数の比較

- 重要な変数を選ぶ

  varImp(cv_rf) # ランダムフォレストの場合

Rank	ランダムフォレスト	PLS-DA	Gradient boosting	Elastic net
1	Phosphatidylcholine (28:2)	Creatine	Phosphatidylcholine (28:2)	Creatinine
2	Ceramide phospho ethanolamine (35:2) Sphingomyelin (32:2)	1,3,7-Trimethyluric acid	Creatinine	Ceramide phospho ethanolamine (35:2) Sphingomyelin (32:2)
3	Creatinine	DL-2-Aminooctanoic acid	Creatine	Phosphatidylcholine (28:2)

上位3位以内の物質で比較すると、重複しているものが多い