判別分析

評価

村田 昇

2020.11.27

講義の予定

- 第 1 日: 判別分析の考え方
- **第 2 日: 判別分析の評価**

判別分析の復習

判別分析

- 個体の特徴量からその個体の属するクラスを予測する関係式を構成
- **事前確率**: $\pi_k = P(Y = k)$ (prior probability)
  - $X = \boldsymbol{x}$ が与えられる前に予測されるクラス
- **事後確率**: $p_k(\boldsymbol{x})$ (posterior probability)
  - $X = \boldsymbol{x}$ が与えられた後に予測されるクラス

$$p_k(\boldsymbol{x}) := P(Y = k|X = \boldsymbol{x})$$

  - 所属する確率が最も高いクラスに個体を分類

判別関数

- 判別の手続き
  - 説明変数 $X = \boldsymbol{x}$ の取得
  - 事後確率 $p_k(\boldsymbol{x})$ の計算
  - 事後確率最大のクラスにデータを分類
- **判別関数**: $\delta_k(\boldsymbol{x})$ $(k = 1, \ldots, K)$

$$p_k(\boldsymbol{x}) < p_l(\boldsymbol{x}) \Leftrightarrow \delta_k(\boldsymbol{x}) < \delta_l(\boldsymbol{x})$$

  事後確率の順序を保存する計算しやすい関数
- 判別関数 $\delta_k(\boldsymbol{x})$ を最大化するようなクラス k に分類

線形判別

- $f_k(\boldsymbol{x})$ の仮定:
  - q 変量正規分布の密度関数
  - 平均ベクトル $\boldsymbol{\mu}_k$: クラスごとに異なる
  - 共分散行列 Σ: **すべてのクラスで共通**

$$f_k(\boldsymbol{x}) = \frac{1}{(2\pi)^{q/2}\sqrt{\det \Sigma}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^{\mathsf{T}}\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right)$$

- 線形判別関数: $\boldsymbol{x}$ の 1 次式

$$\delta_k(\boldsymbol{x}) = \boldsymbol{x}^{\mathsf{T}}\Sigma^{-1}\boldsymbol{\mu}_k - \frac{1}{2}\boldsymbol{\mu}_k^{\mathsf{T}}\Sigma^{-1}\boldsymbol{\mu}_k + \log \pi_k$$

2 次判別

- $f_k(\boldsymbol{x})$ の仮定:
  - q 変量正規分布の密度関数
  - 平均ベクトル $\boldsymbol{\mu}_k$: クラスごとに異なる
  - 共分散行列 Σ_k: **クラスごとに異なる**

$$f_k(\boldsymbol{x}) = \frac{1}{(2\pi)^{q/2}\sqrt{\det \Sigma_k}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^{\mathsf{T}}\Sigma_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right)$$

- 2 次判別関数: $\boldsymbol{x}$ の 2 次式

$$\delta_k(\boldsymbol{x}) = -\frac{1}{2}\det \Sigma_k - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^{\mathsf{T}}\Sigma_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k) + \log \pi_k$$

Fisher の線形判別

- 新しい特徴量 $Z = \boldsymbol{\alpha}^{\mathsf{T}}X$ を考える
- 良い Z の基準:
  - クラス内では集まっているほど良い ($\boldsymbol{\alpha}^{\mathsf{T}}W\boldsymbol{\alpha}$ は小)
  - クラス間では離れているほど良い ($\boldsymbol{\alpha}^{\mathsf{T}}B\boldsymbol{\alpha}$ は大)
- Fisher の基準:

$$\text{maximize} \quad \boldsymbol{\alpha}^{\mathsf{T}}B\boldsymbol{\alpha} \quad \text{s.t.} \quad \boldsymbol{\alpha}^{\mathsf{T}}W\boldsymbol{\alpha} = \text{const.}$$

- $\boldsymbol{\alpha}$ は $W^{-1}B$ の第 1 から第 $K-1$ 固有ベクトル
- 判別方法: 特徴量の距離を用いる
  - $d_k = \sum_{l=1}^{K-1}(\alpha_l^{\mathsf{T}}\boldsymbol{x} - \alpha_l^{\mathsf{T}}\mu_k)^2$ が最小のとなるクラス k に判別

2 値判別分析の評価

誤り率

- 単純な誤り:

$$(\text{誤り率}) = \frac{(\text{誤って判別されたデータ数})}{(\text{全データ数})}$$

- 判別したいラベル: 陽性 (positive)
  - **真陽性**: 正しく陽性と判定 (true positive; TP)
  - **偽陽性**: 誤って陽性と判定 (false positive; FP) (**第 I 種過誤**)
  - **偽陰性**: 誤って陰性と判定 (false negative; FN) (**第 II 種過誤**)
  - **真陰性**: 正しく陰性と判定 (true negative; TN)

混同行列

	真値は陽性	真値は陰性
判別は陽性	真陽性 (True Positive)	偽陽性 (False Positive)
判別は陰性	偽陰性 (False Negative)	真陰性 (True Negative)

- **confusion matrix**
- 転置で書く流儀もあるので注意

混同行列

	判別は陽性	判別は陰性
真値は陽性	真陽性 (True Positive)	偽陰性 (False Negative)
真値は陰性	偽陽性 (False Positive)	真陰性 (True Negative)

- パターン認識や機械学習で多く見られた書き方
- 誤差行列 (error matrix) とも呼ばれる

基本的な評価基準

- 定義

$$\begin{aligned}
(\text{真陽性率}) &= \frac{TP}{TP+FN} && \text{(true positive rate)} \\
(\text{真陰性率}) &= \frac{TN}{FP+TN} && \text{(true negative rate)} \\
(\text{適合率}) &= \frac{TP}{TP+FP} && \text{(precision)} \\
(\text{正答率}) &= \frac{TP+TN}{TP+FP+TN+FN} && \text{(accuracy)}
\end{aligned}$$

- 別名 (分野で異なるので注意)
  - 感度 (sensitivity) あるいは 再現率 (recall):

$$(\text{真陽性率}) = \frac{TP}{TP+FN}$$

  - 特異度 (specificity):

$$(\text{真陰性率}) = \frac{TN}{FP + TN}$$

– 精度:

$$(\text{正答率}) = \frac{TP + TN}{TP + FP + TN + FN}$$

F-値

- **F-measure, F-score**
- 定義

$$F_1 = \frac{2}{1/(\text{再現率}) + 1/(\text{適合率})}$$
$$F_\beta = \frac{\beta^2 + 1}{\beta^2/((\text{再現率}) + 1/(\text{適合率})}$$

再現率 (真陽性率) と適合率の (重み付き) 調和平均

Cohen の kappa 値

- Cohen's **kappa measure**
- 定義

$$p_o = \frac{TP + TN}{TP + FP + TN + FN} \qquad \text{(accuracy)}$$
$$p_e = \frac{TP + FP}{TP + FP + TN + FN} \cdot \frac{TP + FN}{TP + FP + TN + FN} + \frac{FN + TN}{TP + FP + TN + FN} \cdot \frac{FP + TN}{TP + FP + TN + FN}$$
$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e}$$

観測された精度と偶然の精度の比較

実習

データセットの準備

- 以下のデータセットを使用します
  - `winequality-red.`

    UC Irvine Machine Learning Repository で公開されている Wine Quality Data Set の一部

    https://archive.ics.uci.edu/ml/datasets/Wine+Quality
  - 以下に download せずに読み込む方法を紹介します

```
WQ.org <- read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequali
                   sep=";") # データの区切りが ";" となっている
WQ.data <- transform(WQ.org,
                     quality=as.factor( # quality を再分類
```

```
                    ifelse(quality %in% 7:10, "A",
                    ifelse(quality %in% 5:6, "B" ,"C"))))
```

R: 混同行列 caret::confusionMatrix()

- caret: 評価のためのパッケージ
- 判別結果の評価

```
install.packages("caret") # 右下ペインの package タブから install
library(caret) # または require(caret)
confusionMatrix(data, reference)
## data: 判別関数による予測ラベル (factor)
## referenc: 真のラベル (上と同じ factor である必要がある)
```

練習問題

- 前回と同様に東京の機構データの線形判別を行い，以下を確認しなさい
  - 10 月と 11 月の気温と湿度のデータを抽出する

```
TW.data <- transform(read.csv("data/tokyo_weather_reg.csv"),
                     month=substr(as.Date(date),6,7)) # 月を切り出し
TW.subset  <- transform(subset(TW.data,
                               subset= month %in% c("10","11"),
                               select=c(temp,humid,month)),
                        month=as.factor(month)) # 因子にする
```

  - 全てデータを用いて線形判別関数を構成する
  - 構成した判別関数の評価を行う

練習問題

- Wine Quality Data Set を用いて以下を確認しなさい
  - 全てデータを用いて線形判別関数を構成し，評価を行う
  - 全てデータを用いて 2 次判別関数を構成し，評価を行う

予測誤差

訓練誤差と予測誤差

- **訓練誤差**: 既知データに対する誤り (training error)
- **予測誤差**: 未知データに対する誤り (predictive error)
- 訓練誤差は予測誤差より良くなることが多い
- 既知データの判別に特化している可能性がある
  - 過適応 (over-fitting)
  - 過学習 (over-training)

交叉検証

- データを訓練データと試験データに分割して用いる
  - **訓練データ**: 判別関数を構成する (training data)
  - **試験データ**: 予測精度を評価する (test data)
- データの分割に依存して予測誤差の評価が偏る
- 偏りを避けるために複数回分割を行ない評価する
- “交差” と書く場合もある

交叉検証法

- **cross-validation (CV)**
- k-重交叉検証法 (k-fold cross-validation; k-fold CV)
  - n 個のデータを k ブロックにランダムに分割
  - 第 i ブロックを除いた $k-1$ ブロックで判別関数を推定
  - 除いておいた第 i ブロックで予測誤差を評価
  - $i = 1, \ldots, k$ で繰り返し k 個の予測誤差で評価 (平均や分散)
- leave-one-out 法 (leave-one-out CV; LOO-CV)
  - $k = n$ として上記を実行

実習

R: LOO 交叉検証法

- 関数 `lda()` と `qda()` はオプションで LOO 交叉検証を行うことができる
- オプションの指定方法

```
est <- lda(formula, data, CV=TRUE)
est$class # LOO CV による予測結果
## 特定のデータを除いて判別関数を構成し，そのデータの予測を行っている
est <- qda(formula, data, CV=TRUE)
est$class # LOO CV による予測結果
## 2次判別についても同様
```

R: k-重交叉検証法 `caret::train()`

- `caret` パッケージの関数 `train()` で実行可能

```
train(formula, data,
      method,
      trControl=trainControl(method="cv", number))
## formula: R の式
## data: データフレーム
## method: 推定を行う関数 method="lda"/"qda" などを指定
## trControl: 学習方法の指定
## trainControl のオプション
##  method: 評価方法など指定 method="cv"/"LOOCV"
##  number: k-重交叉検証のブロック数 (k)
```

練習問題

- Wine Quality Data Set を用いた線形判別と 2 次判別に関して，以下を行いなさい
  - LOO 交叉検証法を用いて予測誤差の評価を行いなさい
  - k-重交叉検証法を用いて予測誤差の評価を行いなさい

次週の予定

- **第 1 日: クラスター分析と階層的クラスタリング**
- 第 2 日: 非階層的クラスタリング