補論: Ridge/Boosting
機械学習

川田恵介
東京大学
keisukekawata@iss.u-tokyo.ac.jp

2025-12-01

1 他の選択肢

1.1 線型モデルと回帰木モデル

- 機械学習分野において、大量の予測モデル推定方法が提案されている
- 線型モデルと回帰木モデル系統において、他の有力な代替案が存在
  - Stacking を用いれば、予測モデルに容易に組み込める
- ここでは Ridge 法と Boosting 法を紹介
  - 交差推定によるペナルティ項の決定方法も紹介

2 罰則付き回帰

2.1 罰則付き回帰の基本手順

- Step 1. **研究者** が線型モデル $\beta_0 + \beta_1 X_1 + ..$ を設定
- Step 2. 以下を最小化するように、線型モデルのβを算出

$$\textbf{データへの不適合度} + \lambda \times \textbf{複雑性の測定値}$$

- ポイントはλ と複雑性の測定値の決め方

2.2 交差推定による λ の決定

- 交差推定を活用して、λ を選ぶこともできる

1. ある λ のもとで、モデルを交差推定し、平均二乗誤差を計算する

2. 1 を複数のλ について行い、最も予測性能が高い λ を採用する

2.3 Ridge

- LASSO: $\textbf{複雑性の測定値} = |\beta_1| + |\beta_2| ...$

- Ridge: **複雑性の測定値** $= \beta_1^2 + \beta_2^2$
- Ridge の方が母平均をよりよく近似できる状況も存在 (Abadie and Kasy, 2019)

3 Boosting

3.1 Boosting

- Random Forest と並んでよく持ちられる、回帰木モデルのモデル集計方法
  - こちらも大人気の手法
- アイディアは、シンプルすぎるモデルを複雑にしていく

3.2 Boosting 法

1. X, Y 、および初期の予測モデル $g_0(X)$ を指定
   - Y の平均値など
2. Y を予測する”浅い木”を推定し、予測誤差 $R = Y - g_0(X)$ を算出
3. R を予測する”浅い木”を推定し、予測モデル $g(X)$, 予測誤差 $R = Y - g(X)$ を更新
4. 3 を一定回数繰り返し、最終予測モデル $g(X)$ を算出

3.3 性質

- 浅すぎる回帰木モデルからスタートするので、繰り返す回数が少ないと、データからも母平均からも乖離している
- 繰り返すと、データへの適合が改善する
- 繰り返しすぎると、データにほぼ適合するが、予測力は低下する (over-fitting)

3.4 Tuning Parameter

- 繰り返す回数 = 多くし過ぎると、データに完全に(過剰)適合する
  - Random Forest との大きな違い
- よく用いられる Tuninging 方法は、Early Stopping
  - データの一部を検証用に分割し、モデルの検証データへの当てはまりが低下したら、停止

3.5 “ゆっくり学ぶ”

- 一回でデータへの当てはまりを大きく 改善すると、過剰適合する可能性が高まる
- “学習速度”を落とす
  - Regression Tree の分割回数を減らす
  - 予測モデルの更新速度を落とす

- $g(X) = g(X) + \lambda g_0(X)$

3.6 まとめ

- 回帰木は、線型モデルの有力な代替案
  - Random Forest や Boosting などにより、顕著に予測性能が改善しうる
    - Stacking における重要な構成要素
- まだまだ理論的によくわかっていないことが多い(そうです)
  - Causal ML (Chap 9) を参照

3.7 R による実装

- 機械学習のアルゴリズムについて、大量の package が存在
- 共通の”文法”で実装できる”メタ”的 package の利用を推定
  - R: mlr3, tidymodels
  - Python: scikit-learn
- mlr3, scikit-learn は、比較分析用 package である DoubleML の基盤ともなっているので、特におすすめ
  - 本稿では、よりシンプルな SuperLearner を紹介
    - 実装可能な方法が限定的な点に注意

3.8 実装

```
library(tidyverse)
library(SuperLearner)

data("CPS1985", package = "AER")

Y <- CPS1985$wage

X <- select(
  CPS1985,
  education,
  age,
  gender,
  ethnicity)

X <- fastDummies::dummy_cols(
  X,
  remove_selected_columns = TRUE) # ダミー変数に変換
```

3.9 実装

```r
model <- SuperLearner(
  X = X,
  Y = Y,
  SL.library = c(
    "SL.lm",
    "SL.ranger",
    "SL.glmnet",
    "SL.ridge",
    "SL.xgboost"), # 利用したいアルゴリズム
  cvControl = list(V = 2) # 2分割
)

model
```

```
Call:
SuperLearner(Y = Y, X = X, SL.library = c("SL.lm", "SL.ranger", "SL.glmnet",
    "SL.ridge", "SL.xgboost"), cvControl = list(V = 2))

                    Risk      Coef
SL.lm_All       20.03478 0.8920908
SL.ranger_All   21.46856 0.1079092
SL.glmnet_All   20.04239 0.0000000
SL.ridge_All    20.03102 0.0000000
SL.xgboost_All 31.49001 0.0000000
```

3.10 注意

- boosting について、繰り返す回数は初期値を使っている
- early stopping を用いた場合は、mlr3 などの方が簡単

3.11 Reference

Bibliography

Abadie, A. and Kasy, M. (2019) “Choosing among regularized estimators in empirical economics: The risk of machine learning,” Review of Economics and Statistics, 101(5), pp. 743–762.