Linear model for predictions

川田恵介

Table of contents

1 予測問題

1.1 問題の定式化

- 課題: データと同じ母集団からランダムサンプリングされる事例について、X から Y を予測するモデル $g_Y(X)$ をデータから構築する
  - 予測精度は二乗誤差の**母平均** (平均二乗誤差; MSE) で測定

$$E[(Y - g_Y(X))^2]$$

- 母集団外へ拡張可能? (Rothenhäusler and Bühlmann 2023)

1.2 予測精度の推定

- あるモデルの予測精度は母集団上で定義された Estimand
  - データから推定する必要がある
- 代表的なアプローチは、**データ分割**
  - データを Training/Test データにランダム分割し、Training データに対して Algorith を提供し、Test データで予測精度を推定する
    - 80:20, 95:5 などの比率が代表的

1.3 予測精度の指標

- 例: 推定されたモデル $\hat{g}_Y(X)$ について、テストデータから、平均二乗誤差 $E[(Y - \hat{g}_Y(X))^2]$ を推定
  - 決定係数 (R2) $= 1 - (E[(Y - \hat{g}_Y(X))^2]/var(Y))$ はより解釈しやすい

    * $g_y(X)$ が予測した Y の変動

- Linear Model については、伝統的な理論的指標である AIC/BIC も候補

1.4 理想の予測モデル

- $E[(Y - g_Y(X))^2]$ を最小化する予測モデルは母平均 $E[Y|X]$
  - 母平均を Estimand として推定する問題に帰結
    * 事例数が多く、X の数が少なければ、OLS 推定は有力候補
  - $\iff$ OLS は $E[Y|X]$ の (研究者が設定する) 線形近似 (Linear approximation) が Estimand

1.5 一致推定結果

- 無限大の事例数で推定されたモデル $= g_{Y,\infty}(X)$
- 必ずしも Estimand とは一致しない
  - 例: Mis-specification があれば、$g_{Y,\infty}(X) \neq E[Y|X]$

1.6 予測誤差の分解

- $$Y - g_Y(X) = \underbrace{Y - E[Y|X]}_{\text{母集団における個人差: } Irreducible\ error} + \underbrace{E[Y|X] - g_{Y,\infty}(X)}_{\text{母平均と一致推定の乖離: } Approximation\ error} + \underbrace{g_{Y,\infty}(X) - g_Y(X)}_{\text{一致推定と}Estimator\text{の乖離: } Estimation\ error}$$

1.7 例

- $Y \sim \beta_0 + \beta_1 Size$ を 10 事例で推定
- $$\underbrace{Y - g_Y(X)}_{\text{おそらく大きい}} = \underbrace{Y - E[Y|X]}_{Size\text{以外の決定要因があり、おそらく大きい}} + \underbrace{E[Y|X] - g_{Y,\infty}(X)}_{\text{"一直線の関係"ではないので、おそらく大きい}} + \underbrace{g_{Y,\infty}(X) - g_Y(X)}_{\text{事例数が少なすぎ、大きい可能性が高い}}$$

1.8 例

- 事例数が 100 万に増やし、同じモデルを推定する
- $$\begin{aligned} Y - g_Y(X) = & \underbrace{Y - E[Y|X]}_{\text{不変!!!}} \\ & + \underbrace{E[Y|X] - g_{Y,\infty}(X)}_{\text{不変!!!}} \\ & + \underbrace{g_{Y,\infty}(X) - g_Y(X)}_{\text{ほとんど0になることが期待できる}} \end{aligned}$$

1.9 練習問題 (リンク)

- 10 事例のまま、$Y \sim \beta_0 + \beta_1 p(Size, 9)$ を推定した結果、予測性能が大幅に悪化した。何が起こったか?
- $$\begin{aligned} Y - g_Y(X) = & \underbrace{Y - E[Y|X]}_{Irreducible\ Error} \\ & + \underbrace{E[Y|X] - g^*_{Y,\infty}(X)}_{Approximation\ Error} + \underbrace{g^*_{Y,\infty}(X) - g_Y(X)}_{Estimation\ Error} \end{aligned}$$

1.10 例

- 10 事例のまま、$Y \sim \beta_0 + \beta_1 p(Size, 9)$ を推定
- $$\begin{aligned} Y - g_Y(X) = & \underbrace{Y - E[Y|X]}_{\text{不変!!!}} \\ & + \underbrace{E[Y|X] - g_{Y,\infty}(X)}_{\text{減少}} \\ & + \underbrace{g_{Y,\infty}(X) - g_Y(X)}_{\text{非常に大きくなる可能性が高い}} \end{aligned}$$

1.11 まとめ

- モデルを複雑にすると、近似誤差は低下する一方で、推定誤差は増加することが多い
  - Bias-variance トレードオフとして知られる
    - 直感的には、モデルが複雑であれば、より多くをデータに決めさせるので、推定されたモデルはデータの特徴により強く依存する

1.12 まとめ

- 活用できる変数が増えると削減不可能な誤差を減らせる
  - アルゴリズムがうまく扱わないと、予測精度そのものは悪化しうる
- 事例数の増加は、トレードオフを緩和
  - ただし人間が適切にモデルを複雑化する介入が必要
    - 多くの実践で、人間には困難

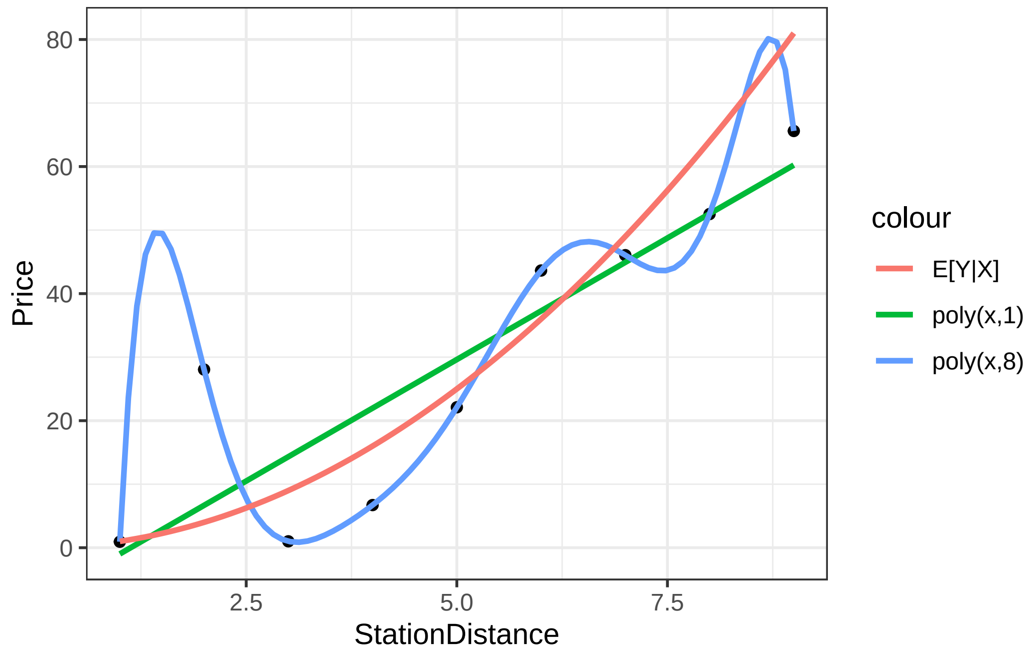
1.13 補論: 過剰適合

- モデルが複雑 (β の数が多い) であれば、推定に用いたデータへの適合度は高くなるが、予測精度は悪化しうる
  - 過剰適合/過学習
- 直感: OLS は $\sum(Y-g_Y(X))^2$ を最小にするように β を決定
  - β の数が増えれば、最小化に用いるフリーパラメタが増えるので、必ず $\sum(Y-g_Y(X))^2$ は減少する

1.14 数値例

```
library(tidyverse)

SimData <- function(n,seed) {
  set.seed(seed)
  Temp <- tibble(
    StationDistance = sample(1:9,n),
    PriceTrue = StationDistance^2,
    Price = PriceTrue + rnorm(n,0,10)
  )
  return(Temp)
}

SimData(9,1) |>
  ggplot(
    aes(
```

```
    x = StationDistance,
    y = Price
  )
) +
theme_bw() +
geom_point() +
geom_smooth(
  aes(
    color = "poly(x,1)"
  ),
  method = "lm",
  se = FALSE
) +
geom_smooth(
  aes(
    color = "poly(x,8)"
  ),
  method = "lm",
  se = FALSE,
  formula = y ~ poly(x,8)
) +
geom_smooth(
  aes(
    y = PriceTrue,
    color = "E[Y|X]"
  ),
  method = "lm",
  se = FALSE,
  formula = y ~ poly(x,8)
)
```



2 Penalized Regression

- 事例数に応じて、適切にモデルの複雑性を調整することは困難
  - X の数が多いと特に難しい
- データ主導で" 自動化" する
  - 代表例は LASSO

2.1 LASSO Algorithm

0. 十分に複雑なモデルからスタート
1. 何らかの基準 (後述) に基づいて Hyper (Tuning) parameter λ を設定
2. 以下の最適化問題を解いて、Linear model $g(X)=\beta_0+\beta_1X_1+\beta_2X_1^2+...$ を推定

$$\min\sum(y_i-g(x_i))^2+\lambda(|\beta_1|+|\beta_2|+..)$$

2.2 Constrained optimization としての書き換え

1. 何らかの基準 (後述) に基づいて Hyper parameter A を設定

2. 以下の最適化問題を解いて、Linear model $g(X)=\beta_0+\beta_1X_1+\beta_2X_1^2+...$ を推定

$$\min\sum(y_i-g(x_i))^2$$

where

$$|\beta_1|+|\beta_2|+..\leq A$$

2.3 λ の役割: OLS

- $\lambda=0$ と設定すれば、(複雑なモデルを)OLS で推定した推定結果と一致
-

$$Y-g_Y(X)=\underbrace{Y-E[Y|X]}_{\text{不変}}$$

$$+\underbrace{E[Y|X]-g_{Y,\infty}(X)}_{\text{小さい}}+\underbrace{g_{Y,\infty}(X)-g_Y(X)}_{\text{大きい傾向}}$$

2.4 練習問題 (リンク)

- λ を極めて大きな値に設定した

1. どのようなモデルになるか?
2. 予測性能が OLS よりも改善した。何が起こったか?

-

$$Y-g_Y(X)=\underbrace{Y-E[Y|X]}_{Irreducible\ Error}$$

$$+\underbrace{E[Y|X]-g^*_{Y,\infty}(X)}_{Approximation\ Error}+\underbrace{g^*_{Y,\infty}(X)-g_Y(X)}_{Estimation\ Error}$$

2.5 λ の役割: 平均

- $\lambda=\infty$ と設定すれば、必ず $\beta_1=\beta_2=..=0$ となる
  - $\beta0$ のみ、最小二乗法で推定: $g(X)=$ サンプル平均
-

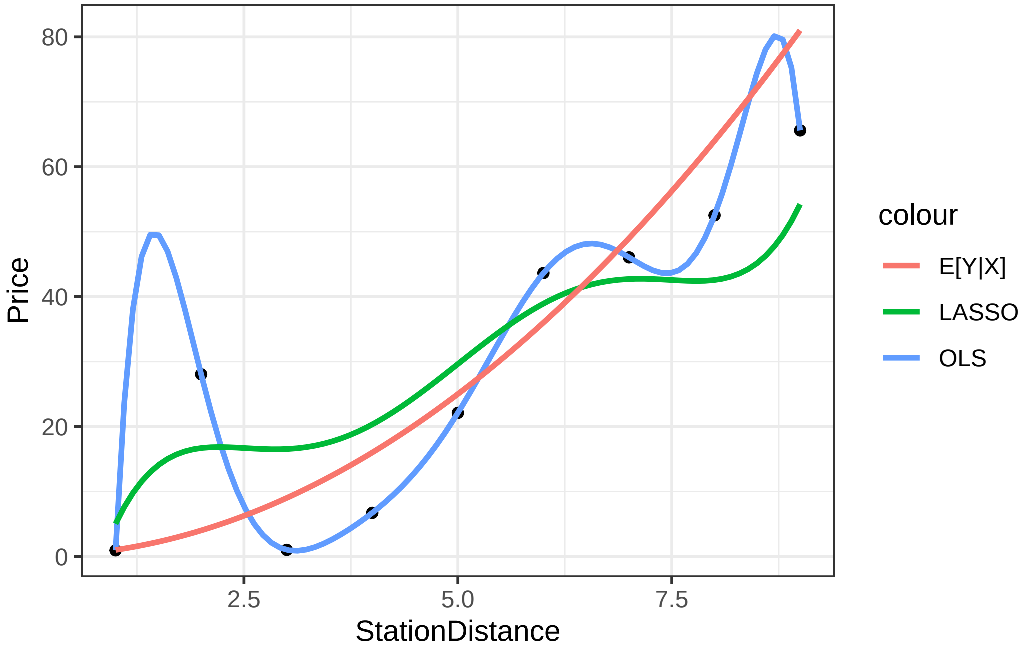
$$Y-g_Y(X)=\underbrace{Y-E[Y|X]}_{\text{不変}}$$

$$+\underbrace{E[Y|X]-g_{Y,\infty}(X)}_{\text{大きい}}+\underbrace{g_{Y,\infty}(X)-g_Y(X)}_{\text{小さい傾向}}$$

2.6 数値例

```r
library(tidyverse)

SimData <- function(n,seed) {
  set.seed(seed)
  Temp <- tibble(
    StationDistance = sample(1:9,n),
    PriceTrue = StationDistance^2,
    Price = PriceTrue + rnorm(n,0,10)
  )
  return(Temp)
}

Y = SimData(9,1)$Price
X = model.matrix(~ 0 + poly(StationDistance,8), SimData(9,1))

OLS = lm(
  Price ~ poly(StationDistance,8),
  SimData(9,1)
  )$fitted

LASSO = gamlr::gamlr(
  x = X,
  y = Y
  ) |>
  predict(X) |>
  as.numeric()

SimData(9,1) |>
  ggplot(
    aes(
      x = StationDistance,
      y = Price
    )
  ) +
  theme_bw() +
  geom_point() +
  geom_smooth(
    aes(
      y = OLS,
```

```
    color = "OLS"
  ),
  method = "lm",
  se = FALSE,
  formula = y ~ poly(x,8)
) +
geom_smooth(
  aes(
    y = LASSO,
    color = "LASSO"
  ),
  method = "lm",
  se = FALSE,
  formula = y ~ poly(x,8)
) +
geom_smooth(
  aes(
    y = PriceTrue,
    color = "E[Y|X]"
  ),
  method = "lm",
  se = FALSE,
  formula = y ~ poly(x,2)
)
```



2.7 λ の役割

- やりたい事: 予測性能を最大化できるように λ を設定し、単純すぎるモデル (Approximation error が大きすぎる) と複雑すぎるモデル (Estimation error が大きすぎる) の間の" ちょうどいい" モデルを構築する
- 設定方法: サンプル分割 (交差推定, glmnet で実装)、情報基準 (gamlr で採用)、理論値 (hdm で採用)
  - 本スライドでは交差推定 (Cross fit/Cross validation) を紹介

3 交差推定

- 訓練/中間評価にどの程度事例を割くか、についてトレードオフ
  - 訓練に事例を割きすぎると、評価が正確にできない
- 交差検証により緩和: 全ての事例を中間評価に持ちいる

3.1 交差推定のアイディア

- 予測性能の高いモデルを算出しやすい λ を使用したい
  - 母平均 $E[Y|X]$ の良い近似モデルを算出しやすい λ を使用したい
- ある λ が生み出すモデルの平均的な予測性能がわかれば、最善の λ を見つけ出せる

- 全く同じ事例を試作と評価に使うと、複雑なモデルを過大評価してしまう
  - 試作と検証は異なる事例でやりさえすれば良い

3.2 シンプルなサンプル分割

- ある λ のもとで推定されるモデルの平均的な性能を評価する

0. データを訓練/中間評価用 (Validation) データに分割
1. 訓練データを用いて、モデルを" 試作" する
2. 中間評価用データを用いて、予測性能を評価する

- 異なる λ について繰り返し、最も性能の良いものを採用

3.3 交差検証

- ある λ のもとで推定されるモデルの平均的な性能を評価する

0. データを細かく分割 (第 1,..,10 サブグループなど)
1. 第 1 サブグループ**以外**で推定して、第 1 サブグループで評価
2. 第 2...サブグループについて、繰り返す
3. 全評価値の平均を最終評価値とする

3.4 数値例: 3 分割

```
library(tidyverse)

SimData <- function(n,seed) {
  set.seed(seed)
  Temp <- tibble(
    StationDistance = sample(1:9,n,replace = TRUE),
    Price = StationDistance + rnorm(n,0,10)
  )
  return(Temp)
}


PopData <- SimData(9,1) |>
```

```r
  mutate(
    Group = sample(rep(1:3,each = 9/3)) |>
      factor()
  )

X = model.matrix(
  ~ 0 + poly(StationDistance,5),
  PopData
)

Y = PopData$Price

FitCV = glmnet::cv.glmnet(
  x = X,
  y = Y
)

PopData
```

```
# A tibble: 9 x 3
  StationDistance    Price Group
            <int>    <dbl> <fct>
1               9   6.05    3
2               4   3.94    2
3               7  31.0     3
4               1   8.64    1
5               2  -5.99    3
6               7  -4.48    1
7               2  -0.895   1
8               3   0.00785 2
9               1  -3.12    2
```

3.5 数値例

- $f_Y(X) = \beta_0 + \beta_1 X + .. + \beta_5 X^5$ を
  - OLS で推定
  - LASSO ($\lambda = 4$) で推定

3.6 数値例: Step 1

```r
Target <- 1

LASSO = glmnet::glmnet(
  x = X[PopData$Group != Target,],
  y = PopData$Price[PopData$Group != Target]
  )

PredOLS <- predict(
  LASSO,
  X,
  s = 0.01
  ) |>
  as.numeric()

PredBest <- predict(
  LASSO,
  X,
  s = 4
  ) |>
  as.numeric()

tibble(
  Price = PopData$Price,
  `Prediction with 4` = PredBest,
  `Prediction with 0.01` = PredOLS,
  SubGroup = PopData$Group
  ) |>
  filter(SubGroup == Target)
```

```
# A tibble: 3 x 4
   Price `Prediction with 4` `Prediction with 0.01` SubGroup
   <dbl>               <dbl>                  <dbl> <fct>
1  8.64               3.92                   -2.76 1
2 -4.48              14.8                    30.3  1
3 -0.895              0.0725                 -5.82 1
```

```
tibble(
  Price = PopData$Price,
  `Prediction with 4` = PredBest,
  `Prediction with 0.01` = PredOLS,
  SubGroup = PopData$Group
  ) |>
  filter(SubGroup != Target)
```

```
# A tibble: 6 x 4
     Price `Prediction with 4` `Prediction with 0.01` SubGroup
     <dbl>               <dbl>                  <dbl> <fct>
1  6.05                 9.03                    6.22  3
2  3.94                 3.61                    4.19  2
3 31.0                 14.8                    30.3   3
4 -5.99                 0.0725                 -5.82  3
5  0.00785              0.462                  -0.228 2
6 -3.12                 3.92                   -2.76  2
```

- Out-of-data R2: -2.57 with 0.01, -0.04 with 4
- In-data R2: 1 with 0.01, 0.53 with 4

3.7 数值例: Step 2

```
Target <- 2

LASSO = glmnet::glmnet(
  x = X[PopData$Group != Target,],
  y = PopData$Price[PopData$Group != Target]
  )

PredOLS <- predict(
  LASSO,
  X,
  s = 0.01
  ) |>
  as.numeric()

PredBest <- predict(
  LASSO,
```

```r
  X,
  s = 4
  ) |>
  as.numeric()

tibble(
  Price = PopData$Price,
  `Prediction with 4` = PredBest,
  `Prediction with 0.01` = PredOLS,
  SubGroup = PopData$Group
  ) |>
  filter(SubGroup == Target)
```

```
# A tibble: 3 x 4
     Price `Prediction with 4` `Prediction with 0.01` SubGroup
     <dbl>                <dbl>                  <dbl> <fct>
1  3.94                  -0.448                  -5.27 2
2  0.00785               21.0                    21.9  2
3 -3.12                   8.18                    8.67 2
```

```r
tibble(
  Price = PopData$Price,
  `Prediction with 4` = PredBest,
  `Prediction with 0.01` = PredOLS,
  SubGroup = PopData$Group
  ) |>
  filter(SubGroup != Target)
```

```
# A tibble: 6 x 4
   Price `Prediction with 4` `Prediction with 0.01` SubGroup
   <dbl>                <dbl>                  <dbl> <fct>
1  6.05                  6.81                   6.06 3
2 31.0                   7.48                  13.2  3
3  8.64                  8.18                   8.67 1
4 -5.99                  2.22                  -3.39 3
5 -4.48                  7.48                  13.2  1
6 -0.895                 2.22                  -3.39 1
```

- Out-of-data R2: -0.84 with 0.01, -0.54 with 4
- In-data R2: 0.16 with 0.01, -0.02 with 4

3.8 数値例: Step 3

```r
Target <- 3

LASSO = glmnet::glmnet(
  x = X[PopData$Group != Target,],
  y = PopData$Price[PopData$Group != Target]
  )

PredOLS <- predict(
  LASSO,
  X,
  s = 0.01
  ) |>
  as.numeric()

PredBest <- predict(
  LASSO,
  X,
  s = 4
  ) |>
  as.numeric()

tibble(
  Price = PopData$Price,
  `Prediction with 4` = PredBest,
  `Prediction with 0.01` = PredOLS,
  SubGroup = PopData$Group
  ) |>
  filter(SubGroup == Target)
```

```
# A tibble: 3 x 4
  Price `Prediction with 4` `Prediction with 0.01` SubGroup
  <dbl>               <dbl>                  <dbl> <fct>
1  6.05               0.683                 -3.50   3
2 31.0                0.683                 -4.44   3
3 -5.99               0.683                 -0.905  3
```

```
tibble(
  Price = PopData$Price,
  `Prediction with 4` = PredBest,
  `Prediction with 0.01` = PredOLS,
  SubGroup = PopData$Group
  ) |>
  filter(SubGroup != Target)
```

```
# A tibble: 6 x 4
     Price `Prediction with 4` `Prediction with 0.01` SubGroup
     <dbl>               <dbl>                  <dbl> <fct>
1  3.94                  0.683                 3.91   2
2  8.64                  0.683                 2.76   1
3 -4.48                  0.683                -4.44   1
4 -0.895                 0.683                -0.905  1
5  0.00785               0.683                 0.0172 2
6 -3.12                  0.683                 2.76   2
```

- Out-of-data R2: -2.6 with 0.01, -1.6 with 4
- In-data R2: 0.91 with 0.01, 0.85 with 4

3.9 比較

- 全データをモデルの試作と中間評価に使用すると、複雑なモデルを過大評価
- データを分割すると、全データを用いた評価はできない
  - 事例数が少ないと評価制精度が悪い
- 交差推定を行えば、過剰適合を避けながら、全データを評価に使用できる
  - 計算時間などの問題点もある

3.10 実践: 単位問題

- LASSO の推定結果は、X の" 単位" に影響を受ける
  - $X = 10\ km/10,000\ m$
  - 実戦では、推定前に平均 0/分散 1 に標準化することが多い
  - 標準化された $X = \frac{X - mean(X)}{var(X)}$

- 「X の一部は Y と強く相関する一方で、相関が弱い変数も大量に存在する」(Approximate Sparsity) 状況で LASSO の予測性能は良好な傾向

3.11 実践: 一致推定量

- 十分に複雑なモデルを設定できれば、LASSO (+ λ のデータ主導の決定)、定式化への依存を減らせる
  - 元々の X について、交差項と連続変数については二乗項を作成など
  - 事例数に応じて λ が減少すれば、母平均の一致推定量を得られる
    - 交差推定など多くの方法で満たされる

3.12 実践: 変数の除外

- LASSO で推定した場合、厳密に 0 となる β が出てくる
  - 非常に稀な場合を除いて、OLS では厳密に 0 にならない (非常に小さいはあり得る)
- $\underbrace{\beta_1}_{=0} \times X_1$ であれば、X_1 をモデルから変数をデータ主導で除外している、と解釈できる
  - Double Selection において重要な手法

3.13 まとめ

- 良い予測には、適度な複雑性を持つモデルが必要
- OLS は人間がモデルを事前に定式化する必要があるが、非常に困難
- ここまでの内容は CausalML Chap 1/3 参照

Reference

Rothenhäusler, Dominik, and Peter Bühlmann. 2023. "Distributionally Robust and Generalizable Inference." *Statistical Science* 38 (4): 527–42.