罰則付き回帰モデル

川田恵介

罰則付き回帰モデル

- X の数が多い場合、線形予測モデルの推定は困難
- 決定木 (RandomForest) は有力な代替案だが、X の数が極めて多くなると機能しなくなる
- 有力な選択肢は、線形予測モデル推定の改良

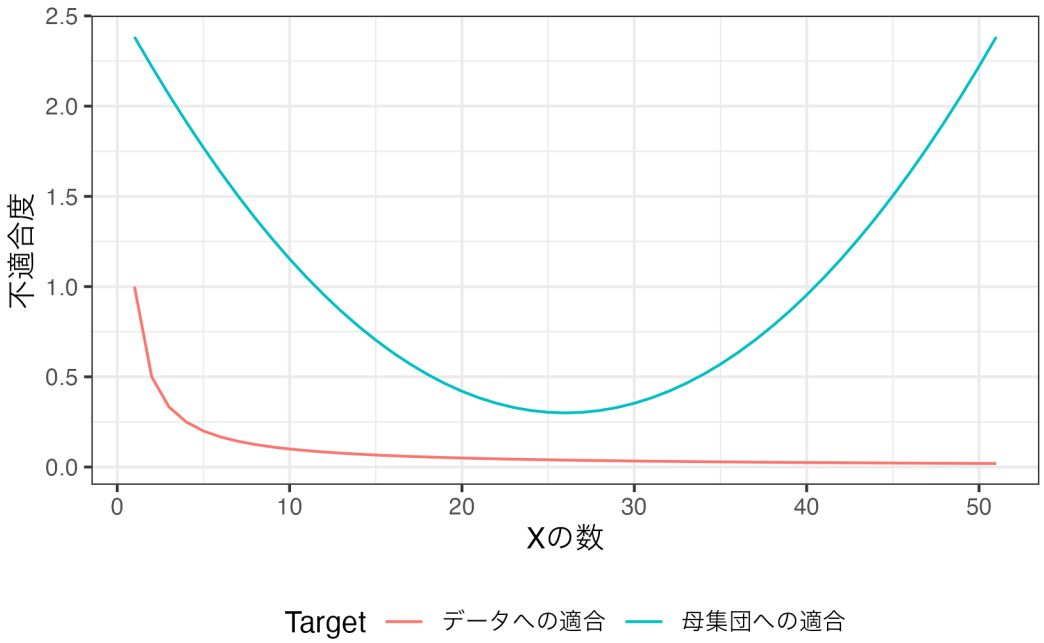
復習

- 線形予測モデル

$$g(X) = \beta_0 + .. + \beta_L X_L$$

- データに当てはめるように推定
  - β の数が多くなると、予測性能が悪化
  - 過剰適合

イメージ



対応

- 環境税などと同じアイディア
- 自動車は便利な道具であるが、同時に排気ガス/渋滞など負の外部性が存在
  - 何も対応しないと保有台数が過大になりうる
- 適切な水準に誘導するために、自動車税を貸す
- **何も対応しないと複雑なモデルになりすぎるので、複雑性に課税する**

罰則付き回帰

- 線形モデル $g(X)$ を、以下を最小化するように推定する

$$\text{データへの当てはまり} + \underbrace{\lambda \times \text{複雑性}}_{\text{複雑性への課税}}$$

- λ：課税額
  - 交差推定で決定
  - 母集団の当てはまり最大化を目指す

複雑性の指標

- Ridge: $\beta_1^2 + .. + \beta_L^2$
- LASSO: $|\beta_1| + .. + |\beta_L|$
- OLS: “0”

LASSO の利点

- 予測において重要ではない β を、厳密に 0 にできる
  - 重要ではない変数をモデルから除外する
- OLS や Ridge では、厳密に 0 にはできない

テキスト分析への有効性

- 単語数が多い $\rightarrow$ X が多い $\rightarrow$ 重要ではない単語も多いかも?
- LASSO が有効な場面も多い

実例

```
36 x 1 sparse Matrix of class "dgCMatrix"
                          s0
(Intercept)     0.39234345
新型コロナ        .
対策             0.14890858
政策             .
日本             .
緊急事態宣言      .
行動            -0.01363740
経済             0.03507508
効果             .
分析             .
危機             .
感染             .
データ          -0.06630760
コロナ           .
ウイルス         .
```

感染拡大	.
影響	.
対応	.
ウィルス感染	.
考察	.
コロナショック	.
pos	0.29916102
みる	.
購買動向	.
消費	.
金融	.
ワクチン接種	.
covid-19	.
比較	.
流行	.
企業	.
雇用	.
ウイルス感染	.
調査	.
変化	.
関係	-0.09086297

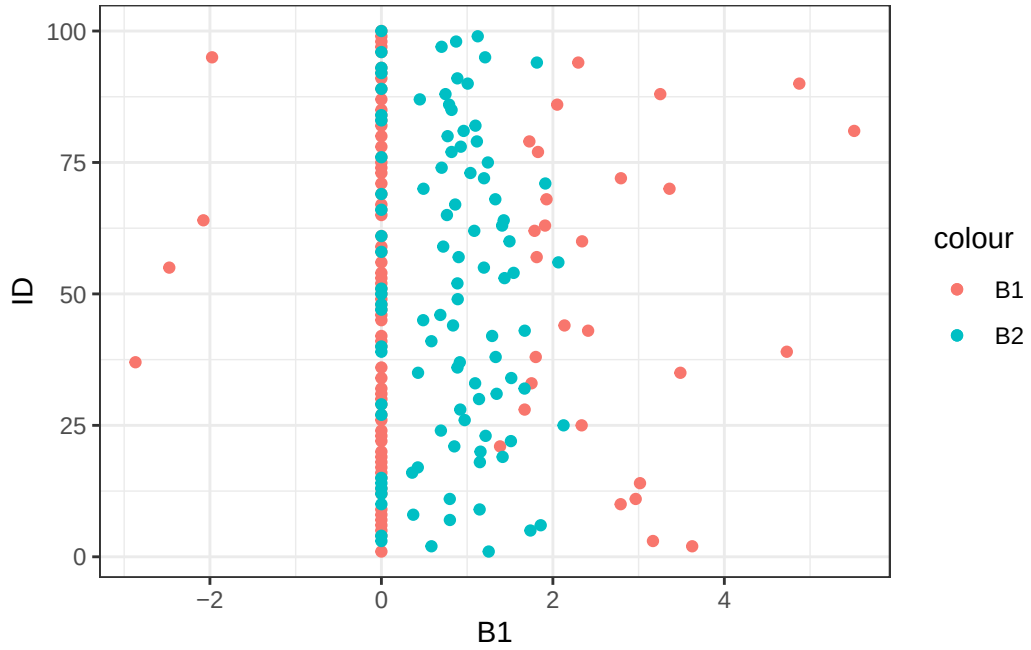
注意: 統計的推論

- 解釈しやすそうなモデルが出てくるが、
  - 母集団の構造への含意は限定的
- Y と関係性が強い変数であったとしても、互いに相関が強ければ、脱落しがち
- 推定誤差の計算は困難

数値例

- $E[Y|X1, X2] = X1 + X2$
- $E[X2|X1] = X1$

数値例



まとめ

- 事例数を大きく超える X から、予測モデルを構築することは困難なチャレンジ
  - 一つのアプローチは、LASSO
  - ただし解釈は慎重に

発展: ニューラルネット

- 線形モデルの拡張
- “人間の脳みその構造を模したモデル”
  - “AI”

アイディア

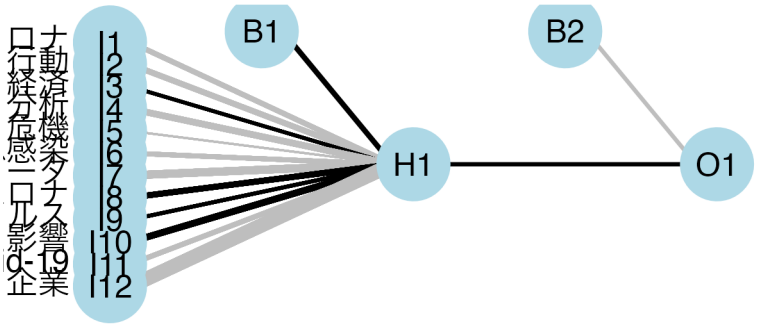
- モデルの集計

1. 複数の中間予測モデル $g_1(X)..g_M(X)$ (Hidden Layer) を推定

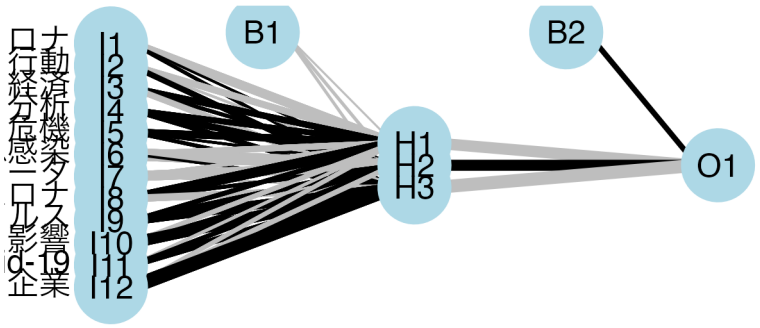
2. 中間予測モデルの予測値から、最終予測モデル $g(g_1(X), .., g_M(X))$ を推定

- g は一般化された線形モデル $g(X_1, .., X_L) = g(\beta_0 + .. + \beta_L X_L)$
- 中間モデルの数 $=$ 複雑性を規定

実例



实例



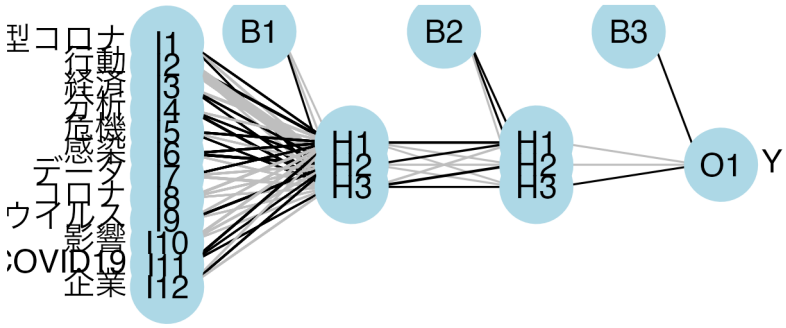
实例

```
Call:
CV.SuperLearner(Y = Y, X = X, SL.library = c("SL.mean", "SL.lm", "SL.glmnet",
    "SL.nnet"))

Risk is based on: Mean Squared Error

All risk estimates are based on V =  10

     Algorithm     Ave        se     Min     Max
 Super Learner 0.23842 0.0121468 0.18934 0.31195
   Discrete SL 0.25600 0.0138381 0.21845 0.38880
   SL.mean_All 0.24435 0.0084204 0.21845 0.28086
     SL.lm_All 0.27505 0.0202366 0.19774 0.39788
 SL.glmnet_All 0.24721 0.0085396 0.22917 0.28086
   SL.nnet_All 0.87968 0.6153254 0.20873 6.30908
```

DeepLearning

- 中間予測モデルを多層にする
- 人間の脳の構造に似ているそうです。。。
  - テキストや画像データについて、高い精度

実例



性能

```
Call:
SuperLearner(Y = Y, X = X, SL.library = c("SL.mean", "SL.lm", "SL.glmnet",
    "SL.nnet"))

                  Risk        Coef
SL.mean_All   0.2466307 0.995873301
SL.lm_All     0.3098022 0.000000000
SL.glmnet_All 0.2514735 0.000000000
SL.nnet_All   1.2454732 0.004126699
```

まとめ

- 盛んに成果が報告される
  - 一部分野では非常に高いパフォーマンス
  - どの程度まで一般性があるかは不透明
- パラメータ設定が難しい