3.5. 入力に対して線形変換を行う方法

入力変数が多く、互いに高い相関を持つ際の手法を考える。
元の入力X_jに対して、少数の線型結合$Z_m(m = 1, \ldots, M)$を計算し、計算結果を入力変数の代わりとして利用する。
この節では、主成分回帰と部分最小2乗法を扱う。

3.5.1 主成分回帰 (Principal Component Regression)

主成分回帰は、以下の流れの繰り返しである。

1. 説明変数のうち、相関のある変数群の線形結合Z_mを作り、これを主成分と呼ぶ。(= **主成分分析**)
2. 主成分のうち、分散の大きい主成分を選ぶ。
3. 累積寄与率が一定値を超えるまで選択し、これを新たな説明変数とする。

入力の主成分を使用していたり、入力変数の大きさに対して不変でないため標準化が必要だったりといった点でリッジ回帰に似ている。

- 主成分分析 ... 相関のある多数の変数の中から、相関のない少数で全体のバラツキをよく表す**主成分**と呼ばれる変数を取り出してくる操作

主成分回帰では、入力列ベクトル$z_m = Xv_m$を計算して得られた$z_1, \ldots, z_M (M < P)$を用いてyの回帰分析を行う。z_mは直交するため、単回帰の総和であると考えられる。

$$\hat{y}^{pcr}_{(M)} = \bar{y}1 + \sum_{m=1}^{M} \hat{\theta}_m z_m \quad (\hat{\theta} = \langle z_m, y \rangle / \langle z_m, z_m \rangle) \tag{3.61}$$

z_mが入力x_jの線形結合なので、(3.61)の解を前述の$\hat{\theta}$及び固有ベクトルv_mを使って、以下の様に表せる。

$$\hat{\beta}(M) = \sum_{m=1}^{M} \hat{\theta}_m v_m \tag{3.62}$$

PCRのメリット

1. 多重共線性の解消 ... 相関のある入力変数を1つの主成分として扱うため、変数間の相関を無くすことができる。
2. **次元削減による過学習の防止** ... 次元数が多いほど、訓練データのノイズにまでフィットしてしまうため。

PCRのデメリット

1. 変数の解釈性の低下 ... 主成分は変数の相関のみを元に和を取るため、意味合いを考えることが難しくなる。
2. **予測性能低下の可能性** ... 主成分が目的変数yを無視して作られるため、必ずしも予測性能が高くなるとは限らない。
3. **情報損失のリスク**

PCRの注意点 ...主成分の数M = 元の変数の数pの際は通常の最小2乗推定と同義。

3.5.2 部分最小2乗法 (Partial Least Squares Regression)

部分最小2乗法は、以下の流れの繰り返しである。

1. 入力$z_1 = \sum_{j=1}^{p} \langle x_j, y \rangle x_j$を求める。
2. 係数$\hat{\theta}_1$を求める。
3. 出力yを係数$\hat{\theta}_1$より求める。
4. z_1について、$x_1, \ldots, x_m$を正規化する。

これを、1からmまで繰り返す。

部分最小2乗法のアルゴリズムは以下の通りである。

1. x_jが平均0、分散1になるように**標準化**をする。
   $\hat{y} = \bar{y}1, x_j^{(0)} = x_j (j = 1, 2, \ldots, p)$
2. 各次元において、**説明変数と目的変数の内積**$\hat{\varphi}_{mj} = \langle x_j^{(m-1)}, y \rangle$を計算する。
3. 計算した$\hat{\varphi}_{mj}$を元に、入力$z_m = \sum_j \hat{\varphi}_{mj} x_j^{(m-1)}$を計算する。
4. 係数$\hat{\theta}_m = \langle z_m, y \rangle / \langle z_m, z_m \rangle$を計算する。
5. 出力$\hat{y}^{(m-1)} = \hat{y}^{(m-1)} + \hat{\theta}_m z_m$を計算する。
6. $x_j^{(m-1)}$をz_mについて直行化する。
   $x_j^{(m)} = x_j^{(m-1)} - [\langle z_m, x_j^{(m-1)} \rangle / \langle z_m, z_m \rangle] z_m (j = 1, 2, \ldots, p)$
7. 2~6を繰り返し、**回帰されたベクトル**$\{\hat{y}^{(m)}\}_{m=1}^{p}$が出力される。
   $\{z_l\}_{l=1}^{m}$は元のx_mに対して線形なので、$\hat{y}^{(m)} = X\beta^{pls}(m)$

PLSのメリット

1. 多重共線性の解消
2. **予測性能** ... 説明変数と目的変数の相関を元に共分散を最大化する潜在変数を選択するため、予測性能が高くなることが多い。

- **潜在変数** ... 直接観測されないが、観測された他の変数から推定される変数

PLSの注意点 ... 主成分の数M $=$元の変数の数pの際は通常の最小2乗推定と同義。

3.6 考察

- 部分最小2乗法、主成分回帰、リッジ回帰は同じような解の傾向を示す。
  lassoはリッジ回帰と最良部分集合選択の中間的な挙動を取る。
- リッジ回帰とlassoは推定の中で、連続的な数値の変化を見せる。
  これに対して、最良部分集合選択や部分最小2乗法、主成分回帰は直線的に数値が変化し、2段階の推定を行っている。

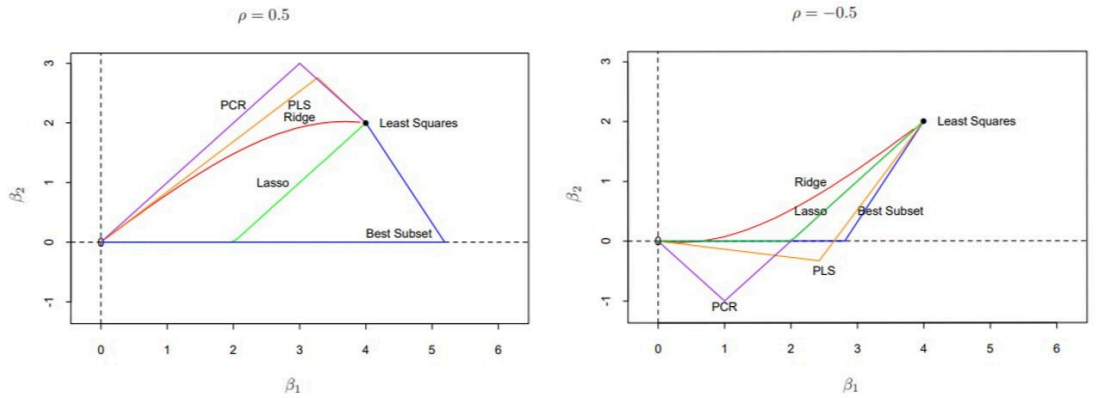

FIGURE 3.18. *Coefficient profiles from different methods for a simple problem: two inputs with correlation ± 0.5, and the true regression coefficients $\beta = (4, 2)$.*

3.7. 複数の目的変数の縮小推定と変数選択 [Munch]

複数の目的変数に対して**変数選択**や**縮小推定**を適用するには、2つの方法がある。

1. 各目的変数に対して独立に適応する方法
2. 全ての目的変数に対して同時に適応する方法

例) リッジ回帰の出力Yの列ベクトルKに対して、

1. 異なるパラメータλを用いると、各目的変数に対して異なる正則化が適応可。
   しかし、k個の正則化パラメータ$\lambda_1, \ldots, \lambda_k$を推測する必要がある。
2. 同じパラメータλを用いると、全ての目的変数に対して同じ正則化となる。
   ただし、k個の目的変数から1つのパラメータλを推測すればよい。

また、複数の目的変数に対して有効な異なる目的変数の相関を利用する方法がある。
以下の様なY_kやY_lが存在するとき、**同じ項$f(X)$を共有しているため**、Y_kやY_lの観測をあわせて利用すれば良いことがわかる。

$$Y_k = f(X) + \epsilon_k$$
$$Y_l = f(X) + \epsilon_l$$

正準相関分析とは、**複数の目的変数の間の相関関係を分析する手法**である。
これにより、共通する成分を持つ変数を抽出できる。

1. x_jの無相関な線形結合$Xv_m(m = 1, \ldots, M)$、目的変数y_kの無相関な線形結合をYu_m作る。
2. 相関$Corr^2(Yu_m, Xv_m)$を最大化する様なv_m、u_mを求める。
3. 2で算出した各v_m、u_mに関して相関が高いほど、変数間に共通があることがわかる。

共通する成分を持つ変数を抽出するメリット

1. **不要な目的変数の削減**
2. ノイズ除去が可
3. **変数間の関連**を予測

縮小ランク回帰とは、複数の目的変数を持つ線形回帰モデルにおいて、変数やモデル・次元を削減する方法である。

1. 前提として、
   p個の独立変数とq個の従属変数が存在。
   $X \in \mathbb{R}^{n \times p}$を中心予測変数、$Y \in \mathbb{R}^{n \times q}$を目的変数とする。
2. **損失関数**$L(B) := \|Y - XB\|^2$を最小化する$B \in \mathbb{R}^{p \times q}$を求める。
3. $rank B \leq r$の制約の下で、$A \in \mathbb{R}^{p \times r}$、$\Gamma \in \mathbb{R}^{r \times q}$より、$B = A\Gamma$として、
   $XB = (XA)\Gamma$と置き換えることで、XAが独立変数見なせる。
   これにより、独立変数がr個に減ったことになる。

縮小ランク回帰のメリット

1. 正則化による**過学習の軽減**
2. 重要な従属変数を選択できる。
   ... XAがXの潜在変数であるため、主成分分析と比べて従属変数を予測する際に重要度の高い変数を選択することができる。

XとYの間の正準変量を上手く縮小することで、平滑化された縮小ランク回帰を考えることができる。

3.8.1. 逐次前向き段階的回帰

逐次前向き段階的回帰は、**ブースティング**や**前向き段階的アルゴリズム**を利用する手法である。

- **ブースティング** ... 大量の弱い学習器の出力を組み合わせることで、強い学習器を生成する方法。
- **前向き段階的アルゴリズム** ... すでに追加された基底関数のパラメータや係数を調整することなく、新たな基底関数を展開に順次追加することにより、損失関数の最小化の式を近似的に解く方法。

逐次前向き段階的回帰のアルゴリズムは以下の通りである。

1. $\beta_1, \beta_2, \ldots, \beta_p = 0$とし、残差$r$が$y$と等しい状態から開始する。
   ただし、説明変数は全て平均0、分散1となるように**標準化**する。
2. 残差rと**最も相関の高い説明変数**x_jを見つける。
3. 係数β_jを$\beta_j \leftarrow \beta_j + \delta_j$に更新する。
   この時、$\delta_j = \epsilon \cdot sign[\langle x_j, r \rangle]$であり、$\epsilon > 0$は小さい更新幅を示す。
   残差rも$r \leftarrow r - \delta_j x_j$と更新する。
4. ステップ2~3を、**残差と説明変数の相関がなくなるまで**繰り返す。

逐次前向き段階的回帰の中で、相関係数の符号と回帰係数の**符号の不一致を解決するため**に、**無限小前向き段階的回帰**(FS $_0$)が考えられた。
以下に最小角回帰を修正したFS $_0$のアルゴリズムを示す。

1. 変数を標準化 (残差$r = y - \bar{y}$、係数$\beta_1, \beta_2, \ldots, \beta_p = 0$)
2. rと最も相関の高いx_jを選択
3. β_jを0から最小2乗係数$\langle x_j, r \rangle$に向かって、他の係数x_kと現在の残差との相関がx_jと同じになるまで、変化させる。
4. **制約付き最小2乗問題を解いて、新しい方向を探索する。**
   s_jは$\langle x_j, r \rangle$の符号である。

$$\min_b \|r - X_{\mathcal{A}} b\|_2^2 \quad subject\ to\ b_j s_j > 0, j \in \mathcal{A}$$

5. p個の変数がモデルに加わるまで繰り返す。

無限小前向き段階的回帰の特徴

1. FS $_0$は、**係数の軌跡が頻繁には方向を変えない**。
2. 係数経路が単調非増加・非減少、及び0をまたがない場合、lassoや最小角回帰と同じ経路を取る。
3. lassoの簡易版と見なせる。

3.8.2. 区分的線形解追跡アルゴリズム

最小角回帰では、lassoの係数軌跡の区分線形性を利用しており、**他の正則化問題にも適応できる**。

例として、以下の様な最適化問題を考える。

$$\hat{\beta}(\lambda) = \underset{\beta}{argmin}[R\beta = \lambda J(\beta)]$$

$$R(\beta) = \sum_{i=1}^{N} L(y_i, \beta_0; \sum_{j=1}^{P} x_{ij}\beta_j)$$

上記の損失関数Lと罰則関数Jが共に凸関数である。

また、以下の2つの条件が係数軌跡$\hat{\beta}(\lambda)$が区分的線形となる十分条件である。

1. Rがβの2次関数か区分的2次関数
2. Jがβについての区分的線形

以上から、係数軌跡が効率的に計算可能であることがわかる。

3.8.3. ダンツィク選択器

lassoの2乗損失誤差を勾配の絶対値で書き換えたものがダンツィク選択器である。

$$\min_{\beta} \|\beta\|_1 \quad subject\ to\ \|X^T(y - X\beta)\|_\infty \leq s$$

lassoと比較した際に、以下の2点で挙動が不安定なことが知られている。

1. 現在の残差と全ての説明変数との内積を最大化しようとするが、残差との相関が低い物をモデルに含めてしまう現象が起きる可能性がある。
2. 正則化パラメータsによっては誤差を多く含む。

3.8.4. グループlasso

ダミー変数(One-Hot encoding)の様なグループの変数が含まれている場合は、**同じグループの変数を選択・係数縮小することが望ましい**。
グループlassoでは、以下の手順でこれを行う。

1. 仮定として、
   p個の変数がL個のグループに分割される。
   p_lをグループlに属する変数の個数である。
   X_lをl番目のグループに属する説明変数、β_lを対応する係数ベクトルとする。
2. 以下の凸基準を最小化する。
   $\sqrt{p_l}$はグループの大きさで、$\|\cdot\|_2$はユークリッドノルムである。

$$\min_{\beta\in\mathbb{R}^p}(\|y-\beta_0 1-\sum_{l=1}^{L}X_l\beta_l\|_2^2+\lambda\sum_{l=1}^{L}\sqrt{p_l}\|\beta_l\|_2) \tag{3.80}$$

3.8.5. lassoの性質について

- 以下の不等式から、良い変数Sと邪魔な変数S^cの相関は高くないことがわかる。
  ただし、Sは真のモデルの非ゼロ係数特徴量の指標で、X_Sが対応する列ベクトルである。
  また、S^cは真の係数が0の変数を示す指標で、X_{S^c}が対応する列ベクトルである。

$$\|({X_S}^T X_S)^{-1} {X_S}^T S_{S^c}\| \leq (1-\epsilon)$$

lassoによる**係数縮小**は非ゼロ係数の推定に対して0に向かうバイアスをかけることになる。
しかし、変数の数が多い時には上手くいかないことがある。
この問題を解決するために、以下の2つの手法がある。

1. 非ゼロ係数の変数の特定にlassoを使い、選ばれた変数に対して**線形モデルを当てはめる**。
   選ばれた変数が多いときは上手くいかない。
2. 非ゼロ係数の変数の特定にlassoを使い、選ばれた変数に対して**lassoを使い係数を求める**。(= **緩和lasso**)
   1度目のlassoと比べると、2度目のlassoはそれほど係数縮小が起きない。

大きな係数に対しては係数縮小をし過ぎないように、**lassoの罰則関数を修正できる**。
平滑打ち切り絶対偏差罰則(SCAD罰則)では、大きな係数βに対する係数の縮小幅を減少させることが可能。
また、aを無限大にすると、係数の縮小効果がなくなる。
非凸関数であるため、**計算が困難**という欠点がある。

$$\frac{dJ_a(\beta,\lambda)}{d\beta} = \lambda \cdot sign(\beta)[I(|\beta| \leq \lambda) + \frac{a\lambda - |\beta|}{(a-1)\lambda} + I(|\beta| > \lambda)] \qquad (3.82)$$

対して、適応lassoは凸性が存在しているうえ、一致性を持った係数推定ができる。
以下の重み付き罰則を使用することで、**変数選択の一致性**と**推定量の一致性**を担保することができ、変数の縮小をさせることが可能。
$\hat{\beta}_j$は通常の最小2乗推定による係数である。

$$\sum_{j=1}^{p} \omega_j |\beta_j| (\omega_j = 1/|\hat{\beta}_j|^{\nu}) \quad (\nu > 0) \tag{3.82}$$

- **変数選択の一致性** ... 非ゼロの変数が正しく選択される確率が1となる性質。
- **推定量の一致性** ... 非ゼロな係数の推定量が真の値に収束する性質。

3.8.6 総当り座標最適化

単純な座標降下法も、lasso解を計算する方法の1つである。
ラグランジュ形式(3.52)の罰則パラメータλを**固定**して、**1パラメータ**のみ**値を変動**させられる状況を考えていき、最適化を行う。

$$\hat{\beta}^{lasso} = \underset{\beta}{argmin}\{\frac{1}{2}\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j) + \lambda\sum_{j=1}^{p}|\beta_j|\} \qquad (3.52)$$

説明変数は平均0、ノルム1の状態に**標準化**されている時、(3.52)は(3.83)のように変形できる。

$$R(\tilde{\beta}(\lambda),\beta)=\frac{1}{2}\sum_{i=1}^{N}(y_i-\sum_{k\neq j}x_{ik}\tilde{\beta}_k(\lambda)-x_{ij}\beta_j)^2+\lambda\sum_{k\neq j}|\tilde{\beta}_k(k)|+\lambda|\beta_j| \quad (3.83)$$

この時、部分残差和$y_i-\tilde{y}_i^{(i)}=y_i-\sum_{k\neq j}x_{ik}\tilde{\beta}_k(\lambda)$を目的変数とした1変数lassoと見なせ、解が明示的に求められる。

$$\tilde{\beta}(\lambda)\leftarrow S(\sum_{i=1}^{N}s_{ij}(y_i-\tilde{\beta}_i^{(j)}),\lambda) \quad (3.84)$$

この時、$S(t,\lambda)=sign(t)(|t|-\lambda)_+$は**ソフト閾値処理**で、λ以下の値は0に落ちる。
係数が収束するまで(3.84)を各変数に適応していくとlasso推定値$\hat{\beta}(\lambda)$が得られる。

これを使って、以下の手順でlasso解を求めることができる。

1. $\hat{\beta}(\lambda_{max}) = 0$となる最小の値$\lambda_{max}$から処理を開始
2. 収束するまで、λを少しずつ小さくしながら、係数の更新を行う。

3.9. 計算上考慮すべき事柄

最小2乗当てはめは、行列X^TXの**コレスキー分解**かXの**QR分解**で計算される。
標本数N、特徴次元数pとすると、

- コレスキー分解の計算量は、$p^3 + Np^2/2$
- QR分解の計算量は、Np^2
  であるため、コレスキー分解のほうが計算量は少ないが、数値的には不安定である。
  また、最小角回帰のアルゴリズムを用いて、lassoの数値解を計算計算する時の計算量は、最小2乗回帰と同じオーダーである。

参考文献

[1]主成分分析 - Wikipedia (参照:2025/05/08)
https://ja.wikipedia.org/wiki/主成分分析

[2]次元削減手法（まとめと実装）PCA, LSI(SVD), LDA, ICA, PLIS #機械学習 - Qiita (参照:2025/05/11)
https://qiita.com/Hatomugi/items/d6c8bb1a049d3a84feaa

[3]部分的最小二乗回帰 - Wikipedia (参照:2025/05/14)
https://ja.wikipedia.org/wiki/部分的最小二乗回帰

[4]正準相関分析（Canonical Correlation Analysis: CCA） #Python - Qiita
(参照:2025/05/14)
https://qiita.com/yoneda88/items/847cb99542538083b876

[5]「縮減順位回帰」とは一体何なのか？ - Cross Validated (参照:2025/05/14)
https://stats.stackexchange.com/questions/152517/what-is-reduced-rank-regression-all-about

[6] スパースモデリング（応用編）- ごちきか (参照:2025/05/13)
https://gochikika.ntt.com/Modeling/regularization_advanced.html