基数制約付きLTS回帰問題に対する DC最適化アルゴリズム

A DC optimization algorithm

for cardinality-constrained LTS regression problem

経営システム工学専攻
加藤　桃

1. 序論

データ分析によく用いられる手法の一つに回帰分析がある．回帰分析において，説明変数の候補から有用な説明変数を選ぶ変数選択は重要な課題の一つである．説明変数を選択することにより回帰式の結果の解釈が容易になったり過剰適合を防ぐことができたりするため，未知データに対する予測精度も向上する [2].

変数選択の方法の一つに，係数ベクトルのノルムをペナルティ項として加える正則化と呼ばれる手法が注目されている．

また,「使用する変数の数を高々K 個以下に抑える」という観点から行う，基数制約という手法も提案されている [1].

統計解析において，得られたデータの中に外れ値が含まれることが多々ある．外れ値は少量であったとしても回帰式の推定に大きく影響を与えてしまうため，外れ値の取り扱いについては以前から広く研究が行われている．外れ値を考慮した回帰分析の一つとして Least Trimmed Squares（最小2乗刈込平均最小化）（LTS 回帰）がある [3]．LTS 回帰とは，残差平方を昇順に並び替えた統計量のκ 番目までの和を最小化する手法である．しかし，LTS 回帰に対して基数制約の観点から変数選択を行う手法は少ない．

そこで本研究では，Gotoh et al.[1] による DC 表現を適用した基数制約付き LTS 回帰を提案する．近接写像を利用する近接 DC アルゴリズムを適用することで，効率的に停留点を求めることができる利点がある．

数値実験では，データの性質の違いによる DC アルゴリズムの振る舞いの違いについて示し，未知データに対する予測精度（目的関数の平均値）の検証結果を示す．

2. 定式化

LTS 回帰の定式化および基数制約を組み込んだ LTS 回帰の定式化についてまとめる．

2.1. LTS 回帰の定式化

まず，線形回帰モデルに対する通常の推定方法から簡単に説明する．線形回帰モデルは，p 個の説明変数 $x_1, ..., x_p$ と被説明変数 y の関係を，定数項 β_0 と回帰係数 $\beta_1, ..., \beta_p$，残差項 ε を用いて，$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon$ と仮定する．説明変数の定数項と回帰係数 $\bar{\boldsymbol{\beta}} = (\beta_0, \beta_1, ..., \beta_p)^\top$，$\bar{\boldsymbol{X}}_i = (1, x_{i1}, ..., x_{ip})^\top$ とすると，$\bar{\boldsymbol{\beta}}$ は以下の最小化問題を解くことで求める．

$$\underset{\bar{\boldsymbol{\beta}}}{\text{minimize}} \sum_{i=1}^{n} r_i^2(\bar{\boldsymbol{\beta}})$$

ただし，${r_i}^2(\bar{\boldsymbol{\beta}}) = \left(y_i - {\bar{\boldsymbol{X}}_i}^\top \bar{\boldsymbol{\beta}}\right)^2$ とする．

LTS 回帰は，2 乗誤差の累積和ではなく，2 乗誤差が小さい方から κ 個を足し合わせたものを最小化する手法である．説明変数の回帰係数 $\bar{\boldsymbol{\beta}}$ は，以下の最小化問題を解くことで求める．

$$\underset{\bar{\boldsymbol{\beta}}}{\text{minimize}} \sum_{i=1}^{\kappa} r_{[i]}^2(\bar{\boldsymbol{\beta}}) \tag{2.1}$$

ただし，$\kappa \leq n$ とし，$r_{[i]}^2(\bar{\boldsymbol{\beta}})$ は

$$r_{[1]}^2(\bar{\boldsymbol{\beta}}) \ \leq \ r_{[2]}^2(\bar{\boldsymbol{\beta}}) \ ... \leq \ r_{[\kappa]}^2(\bar{\boldsymbol{\beta}}) \ ... \leq \ r_{[n]}^2(\bar{\boldsymbol{\beta}})$$

残差を昇順に並び替えたものである．

2.2. 基数制約付き LTS 回帰の定式化

本研究では，[1] の枠組みを使い，基数制約と LTS 回帰の目的関数を 2 つの凸関数の差（Difference of two Convex functions ; DC）を用いて表現する．

ここで，$\boldsymbol{\beta} \in \mathbb{R}^n$ の非ゼロ要素数を $\|\boldsymbol{\beta}\|_0$ で表す:

$$\|\boldsymbol{\beta}\|_0 := |\{j \in \{1, ..., n\} : \beta_j \neq 0\}|.$$

便宜上，これを $\boldsymbol{\beta}$ の ℓ_0 ノルムと呼ぶ．ℓ_0 ノルムを使うと基数制約は $\|\boldsymbol{\beta}\|_0 \leq K$ と表せる．以下に定義する largest-K ノルムを用いることで，基数制約 $\|\boldsymbol{\beta}\|_0 \leq K$ は DC 表現できる．

定義 1 (largest-K ノルム) ベクトル $\boldsymbol{\beta} \in \mathbb{R}^p$ の要素を絶対値順に並べたときの，上位 K 個の要素の絶対値和を

$$\|\boldsymbol{\beta}\|_K := |\beta_{(1)}| + |\beta_{(2)}| + ... + |\beta_{(K)}|$$

とする [1].

命題 1 $\|\boldsymbol{\beta}\|_0 \leq K$ と，$\|\boldsymbol{\beta}\|_1 - \|\boldsymbol{\beta}\|_K = 0$ が成り立つことは必要十分条件である．ここで $\|\boldsymbol{\beta}\|_1$ は $\boldsymbol{\beta}$ の ℓ_1 ノルムを表す.

命題2.1 より，基数制約 $\|\boldsymbol{\beta}\|_0 \leq K$ は ℓ_1 と largest-K ノルムの差で表すことができる．

また，LTS 回帰の目的関数も ℓ_1 ノルムと largest-K ノルムの差で表すことができる．目的関数は，$r^2_{[i]}$ が少ない方から κ 個までの和を表しているので，総和から，$r^2_{[i]}$ が大きい方から $n-\kappa$ 個までの和（図 2.1 の網掛け部分）を引いた

$$\sum_{i=1}^{\kappa} r^2_{[i]}(\bar{\boldsymbol{\beta}}) = \|\boldsymbol{r}^2(\bar{\boldsymbol{\beta}})\|_1 - \|\boldsymbol{r}^2(\bar{\boldsymbol{\beta}})\|_{n-\kappa}$$

と変形することができる．

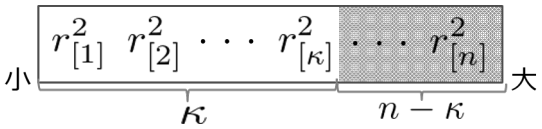

図 2.1: 目的関数の DC 表現

さらに，制約式をペナルティとして 2.1 式に加えることで，基数制約付き LTS 回帰は以下の DC 計画問題として定式化できる：

$$\begin{aligned}\underset{\bar{\boldsymbol{\beta}}}{\text{minimize}}\ f(\bar{\boldsymbol{\beta}}) := \|\boldsymbol{r}^2(\bar{\boldsymbol{\beta}})\|_1 - \|\boldsymbol{r}^2(\bar{\boldsymbol{\beta}})\|_{n-\kappa} \\ +\rho(\|\boldsymbol{\beta}\|_1 - \|\boldsymbol{\beta}\|_K)\end{aligned} \tag{2.2}$$

ただし，ρ はペナルティの強さを表す正の定数である．

3. 基数制約付き LTS 回帰に対する近接 DC アルゴリズム

DC 計画問題に対しては DC アルゴリズムを用いることで，停留点が得られる．一般に DC アルゴリズムでは目的関数（凸関数 $-$凸関数）の$-$凸関数の部分を線形近似し，目的関数を凸関数に近似した問題を解く．線形近似と凸計画問題の求解を繰り返して停留点を求める [4].

2.2 式に対して DC アルゴリズムを適用することを考える．ここで，$\boldsymbol{Q} = \bar{\boldsymbol{X}}^\top \bar{\boldsymbol{X}}$, $\boldsymbol{q} = -\bar{\boldsymbol{X}}^\top \boldsymbol{y}$, $\bar{\boldsymbol{I}}$: 回帰に用いない $n-\kappa$ 個の添え字集合，$\varepsilon > 0$ に対して $\bar{\lambda} := \boldsymbol{Q}$ の最大固有値 $+\varepsilon$ と置き，特殊な変形をすると $f(\bar{\boldsymbol{\beta}})$ は以下のように 2 つの凸関数の差に分解することができる：

$$\begin{aligned}f(\bar{\boldsymbol{\beta}}) = \underbrace{\boldsymbol{y}^\top \boldsymbol{y} + \bar{\lambda}\|\bar{\boldsymbol{\beta}}\|_2^2 + \rho\|\boldsymbol{\beta}\|_1}_{\text{凸部分 1}} \\ - \underbrace{(\bar{\boldsymbol{\beta}}^\top(\bar{\lambda}\boldsymbol{I} - \boldsymbol{Q})\bar{\boldsymbol{\beta}} - 2\boldsymbol{q}^\top\bar{\boldsymbol{\beta}} + \bar{\boldsymbol{\beta}}^\top\boldsymbol{Q}_{\bar{\boldsymbol{I}}}\bar{\boldsymbol{\beta}} + 2\boldsymbol{q}_{\bar{\boldsymbol{I}}}^\top\bar{\boldsymbol{\beta}} + \rho\|\boldsymbol{\beta}\|_K)}_{\text{凸部分 2}}\end{aligned} \tag{3.1}$$

ここで，$\boldsymbol{I}$ は単位行列であり，$\|\cdot\|_2$ は ℓ_2 ノルムである．

$\boldsymbol{\sigma}^{t-1}$ を第 $t-1$ 回目の反復で得られた解 $\bar{\boldsymbol{\beta}}^{t-1}$ による，凸部分 2 の劣勾配としたとき，$\boldsymbol{\sigma}^{t-1}$ は

$$\boldsymbol{\sigma}^{t-1} \in \{2\bar{\lambda}\bar{\boldsymbol{\beta}}^{t-1} - 2(\boldsymbol{Q}-\boldsymbol{Q}_{\boldsymbol{I}})\bar{\boldsymbol{\beta}}^{t-1} - 2(\boldsymbol{q}-\boldsymbol{q}_{\boldsymbol{I}})\} + \rho\cdot\partial\|\boldsymbol{\beta}^{t-1}\|_K$$

で与えられる．

3.1 式に対する DC アルゴリズムでは，第 t 回反復において以下の凸計画問題を解く：

$$\min \quad \boldsymbol{y}^\top\boldsymbol{y} + \bar{\lambda}\|\bar{\boldsymbol{\beta}}\|_2^2 + \rho\|\boldsymbol{\beta}\|_1 - \bar{\boldsymbol{\beta}}^\top\boldsymbol{\sigma}^{t-1} \tag{3.2}$$

ここで，3.2 式は各説明変数ごとの部分問題に分解することができ，3.2 式の最適解を $\bar{\boldsymbol{\beta}}^t = (\beta_0^t, \beta_1^t, ..., \beta_p^t)$ としたとき，β_0^t は

$$\underset{\beta_0}{\text{argmin}} \quad \left\{\left(\beta_0 - \frac{\sigma_0^{t-1}}{2\bar{\lambda}}\right)^2\right\} \tag{3.3}$$

と一致し，$\boldsymbol{\beta}_j^t (j = 1, ..., p)$ はそれぞれ

$$\underset{\beta_j}{\text{argmin}} \quad \left\{\left\|\beta_j - \frac{\sigma_j^{t-1}}{2\bar{\lambda}}\right\|_2^2 + \frac{\rho}{\bar{\lambda}}\beta_j\right\} \tag{3.4}$$

と一致する．近接写像を利用して，以下のように解析解が得られる．

$$\beta_0^t = \frac{\sigma_j^{t-1}}{2\bar{\lambda}}$$

$$\beta_j^t = \begin{cases} \dfrac{\sigma_j^{t-1} - \rho}{2\bar{\lambda}} & (\sigma_j^{t-1} \geq \rho) \\ 0 & (-\rho \leq \sigma_j^{t-1} \leq \rho) \\ \dfrac{\sigma_j^{t-1} + \rho}{2\bar{\lambda}} & (\sigma_j^{t-1} \leq \rho) \end{cases} \quad (j = 1, ..., p)$$

以上をまとめて，3.1 式に対する近接 DC アルゴリズムの流れを以下に示す．

Algorithm 1 近接 DC アルゴリズム

Require: $\bar{\boldsymbol{\beta}}^0, \varepsilon > 0$

$t = 1$

repeat

劣勾配 $\boldsymbol{\sigma}^{t-1}$ を計算する．

$\bar{\beta_0}$ について，3.3 式を計算する．

各 $\bar{\beta_j}$ について，3.4 式を計算する．

$t \leftarrow t + 1$

until $|f^{t-1} - f^t| < \varepsilon$

4. 数値実験

数値実験を行い，データの性質の違いによる DC アルゴリズムの振る舞いの比較と，未知データに対する予測精度（目的関数の平均値）の検証をした．

4.1. 振る舞いの比較

人工データ（データ数 n，説明変数候補数 $p = 100$）による数値実験を行う．説明変数と被説明変数の関係は線形とした．ベクトル $\boldsymbol{X}_i$ をデータ i の説明変数ベクトルとして，$\boldsymbol{X}_i \sim N(0, 10)$ より生成し，被説明変数 y_i は次のように作成した：$y_i = \sum_{j=1}^{p} \beta_j x_{ij} + \beta_0 + \varepsilon_i$

また，外れ値の割合：l，サンプルサイズ：n，意味のある説明変数：q，外し度：d とし，それぞれの項目で 2 通りの値を与えてデータを生成した．

表 4.1: 人工データの変化項目

	l	n	q	d
a	1%	5000	5	0.9
b	3%	50000	50	0.1

説明変数の係数は以下の値を与えた．

$\bar{\boldsymbol{\beta}} = (0, 1, ..., 5, 0, 0, ..., 0)^\top$ （$q = 5$ のとき）

$\bar{\boldsymbol{\beta}} = (0, 1, ..., 50, 0, 0, ..., 0)^\top$ （$q = 50$ のとき）

外れ値は ε_i の違い値によって作成した．通常の y_i（$i = 1, ..., \kappa$）には $\varepsilon_i \sim N(0, 1)$ を与え，外れ値の y_i（$i = \kappa + 1, ..., n$）には $\varepsilon_i \sim N(S, 1)$ を与えた．

S は以下の式で求めた．

$$S = d \cdot 10 \sqrt{\sum_{j=1}^{p} \beta_j^2 + 1}$$

表 4.2: 外れ値の割合の違いによる計算時間の比較

			$\kappa = n$			
			$K = K^*$		$K = 100$	
			外れ値の割合 l			
n	q	d	1%	3%	1%	3%
5000	5	0.9	0.23	0.38	0.019	0.020
		0.1	0.11	0.21	0.018	0.027
	50	0.9	0.60	0.70	0.019	0.024
		0.1	1.49	1.38	0.27	0.025
50000	5	0.9	0.46	3.20	0.28	0.27
		0.1	1.50	1.38	0.27	0.17
	50	0.9	0.67	5.27	0.17	0.17
		0.1	2.47	2.24	0.17	0.17

$K = 100$ のときと $K = K^*$ のときを比較すると，目的関数の平均は大きく変わっておらず，外れ値の割合が多くなってもきちんと意味のある変数を選択できているといえる．また，外れ値の割合が多い方が計算時間が多くかかることが分かった．

表 4.3: 説明変数の違いによる計算時間の比較

			$\kappa = n$			
			$K = K^*$		$K = 100$	
			説明変数 q			
l	n	d	5	50	5	50
1%	5000	0.9	0.23	0.60	0.019	0.019
		0.1	0.11	0.20	0.018	0.021
	50000	0.9	0.46	0.67	0.28	0.17
		0.1	1.50	2.47	0.27	0.17
3%	5000	0.9	0.38	0.70	0.020	0.024
		0.1	0.21	0.39	0.027	0.025
	50000	0.9	3.20	5.27	0.27	0.17
		0.1	1.38	2.24	0.17	0.17

意味のある説明変数が多い方が計算時間も多くかかっていた．$K = 100$ のときと $K = K^*$ のときを比較すると，目的関数の平均は大きく変わっておらず，意味のある説明変数の値が増えてもきちんと意味のある変数を選択できているといえる．

表 4.4: 外し度の違いによる計算時間の比較

			$\kappa = \kappa^*$	
			$K = K(\text{ave})$	
			外れ値の外し度 d	
l	n	q	0.9	0.1
1%	5000	5	0.60	0.31
		50	0.90	0.77
	50000	5	7.92	5.18
		50	11.17	9.43
3%	5000	5	0.70	0.49
		50	1.06	0.83
	50000	5	8.96	7.18
		50	12.22	10.01

外し度が大きい方が計算時間は多くかかっていた．初期点 $\bar{\boldsymbol{\beta}}$ として OLS 解を与えたことが，外し度が大きい方が時間がかかっている理由として考えられる．

4.2. 予測精度の検証

使用したデータは UCI Macine Learning Repository [5] で公開されているデータと人工データを用いた．また今回は，予測精度を目的関数の平均値とし，検証は 5 分割交差検証法により行った．K は 5 分割交差検証法により，κ は AIC 基準によりパラメータを決定した．OLS でチューニングしたパラメータを用いた予測精度との比較を以下に示す．

表 4.5: 検証結果

データ名	DCA	OLS
Forest Fires	88.31	88.32
Housing	43.13	42.93
人工データ aaaa	5.58	5.70
人工データ aaab	1.05	1.08
人工データ aaba	3475.79	3519.83

表 4.2 の通り，Forest Fires のデータと人工データに関しては，提案手法の方が目的関数の平均値が小さくなった．一番大きな差が出たのは人工データ C であった．説明変数の数が多いデータの方が大きな差が出る結果となった．

5. 結論

本研究では，基数制約付き LTS 回帰に対して [1] をもとに，基数制約と LTS 回帰の目的関数を ℓ_1 ノルムと largest-K ノルムの差で表現し，DC アルゴリズムの各反復時に近接写像を利用する近接 DC アルゴリズムを用いた解法を提案した．数値実験では，データの性質の違いによる DC アルゴリズムの振る舞いを分析した．今回試した人工データでは，サンプルサイズが 50000，説明変数が 100 あるデータに対しても，比較的高速に停留点を求めることができた．目的関数の平均値と計算時間の観点から，DC アルゴリズムの適用は有効であることが分かった．

今後の課題としては，DC アルゴリズムにおいてより良い解が得られるためのパラメータの設定方法を探ることが挙げられる．本論文では κ のチューニングに AIC 基準法を適用したが，より良い基準を探す必要がある．

参考文献

[1] J. Gotoh, A. Takeda, and K. Tono, “DC formulation and algorithms for sparse optimization problems,” Mathematical Programming, 2017.

[2] I. Guyon and A. Elisseeff, “An introduction to variable and future selection,” Journal of Machine Learning Research.3, pp.1157–1182, 2003.

[3] J. Rousseeuw, “Least Median of Spuares Regression,” Journal of the American Statistical Association, 1984.

[4] T. Pham Dinh and H. A. Le Thi, “Convex analysis approach to d.c. programming: theory, algorithms and applications,” Acta Mathematica Vietnamica, 22(1), pp.289–355, 1997.

[5] M. Lichman, UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science, 2013. URL: `http://archive.ics.uci.edu/ml`.