Using of Ellipsoid Method for Finding Linear Regression Parameters with L_1-regularization

P. Stetsyuk, **V. Stovba**, M. Korablov

V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine

2nd International Conference on Information Technologies and Their Applications
(ITTA-2024)

April 23-25, 2024

Content

1. Criterion of Least Moduli powered to $1 \le p \le 2$
2. The emlmpr algorithm with L_1-regularization
3. Computational experiments without regularization
4. Computational experiments with regularization

Content

Linear regression and LMP criterion

m observations $(\mathbf{a}_i, y_i)$, $\mathbf{a}_i = (a_{i1}, \dots, a_{in}) \in \mathbb{R}^n$, $y_i \in \mathbb{R}$, $i = \overline{1, m}$

n unknown parameters $x_1, \dots, x_n$ to be evaluated, $m > n$

$$y_i = \sum_{j=1}^{n} a_{ij} x_j + \varepsilon_i, \quad i = \overline{1, m} \tag{1}$$

Criterion of Least Moduli powered to $1 \le p \le 2$ (LMP) is a mathematical programming problem

$$f_p^* = f_p(x_p^*) = \min_{x \in \mathbb{R}^n} \left\{ f_p(x) = \sum_{i=1}^{m} \left| y_i - \sum_{j=1}^{n} a_{ij} x_j \right|^p \right\} \tag{2}$$

LMP with $p = 1, 2$

Least Moduli Method is the problem (2) with $p = 1$

$$f_1^* = \min_{x \in \mathbb{R}^n} \left\{ f_1(x) = \sum_{i=1}^{m} \left| y_i - \sum_{j=1}^{n} a_{ij} x_j \right| \right\} \quad (3)$$

Least Square Method is the problem (2) with $p = 2$

$$f_2^* = \min_{x \in \mathbb{R}^n} \left\{ f_2(x) = \sum_{i=1}^{m} \left(y_i - \sum_{j=1}^{n} a_{ij} x_j \right)^2 \right\} \quad (4)$$

LMP with L_1-regularization

corresponds to the following mathematical programming problem

$$f_p^* = f_p(x_p^*) = \min_{x \in \mathbb{R}^n} \left\{ f_p(x) = \sum_{i=1}^{m} \left| y_i - \sum_{j=1}^{n} a_{ij} x_j \right|^p + \lambda \sum_{j=1}^{n} |x_j| \right\} \quad (5)$$

- Problem (5) is a problem of unconditional minimization of a convex nonsmooth function $f_p(x)$ if $p > 1$ and convex piecewise-linear function if $p = 1$
- $\lambda \geq 0$ is a regularization parameter

Content

The emlmpr algorithm based on ellipsoid method

Input parameters: $B_0 := I_n,\ x_0,\ r_0,\ \varepsilon_f > 0$

Step 1. Calculate $f_p(x_k)$ and $g_{f_p}(x_k)$. If $r_k \left\| B_k^T g_{f_p}(x_k) \right\| \le \varepsilon_f$, then STOP: $k^* = k$ and $x_p^* = x_k$. Otherwise move to Step 2.

Step 2. Let $\xi_k := \frac{B_k^T g_{f_p}(x_k)}{\left\| B_k^T g_{f_p}(x_k) \right\|}$.

Step 3. Calculate the next point

$$x_{k+1} := x_k - h_k B_k \xi_k, \text{ where } h_k = \frac{1}{n+1} r_k.$$

Step 4. Calculate

$$B_{k+1} := B_k + \left(\sqrt{\frac{n-1}{n+1}} - 1 \right) (B_k \xi_k) \xi_k^T \text{ and } r_{k+1} := r_k \frac{n}{\sqrt{n^2 - 1}}.$$

Step 5. Move to iteration $k+1$ with values $x_{k+1},\ r_{k+1},\ B_{k+1}$.

Computational capabilities of emlmpr

Theorem. Sequence of points $\{x_k\}_{k=0}^{k^*}$ satisfy the following inequalities:

$$\left\|B_k^{-1}(x_k - x_p^*)\right\| \leq r_k, \quad k = 0, 1, \ldots, k^*.$$

On each iteration $k > 0$ the value of decreasing of volume of the ellipsoid $E_k = \left\{x \in \mathbb{R}^n : \left\|B_k^{-1}(x_k - x)\right\| \leq r_k\right\}$, which localizes point x_p^*, is constant and equal to

$$q = \frac{vol(E_k)}{vol(E_{k-1})} = \sqrt{\frac{n-1}{n+1}}\left(\frac{n}{\sqrt{n^2-1}}\right)^n < exp\left\{-\frac{1}{2(n+1)}\right\} < 1.$$

The **emlmpr** algorithm can be successfully run on modern computers if $n = 10 \div 30$ and $m = 100 \div 1000$

Content

Computational experiments details

- **3 test examples**
  - solving time estimation
  - robustness of LMP with $p = 1$
  - L_1-regularization effect
- Intel Core i7-10750H, 2.6 GHz, 16 Gb RAM
- Windows 10/64, Octave 6.3.0

Test example 1: solving time estimation

- $\mathbf{n} = 20$, $\mathbf{m} = 200$
- $\mathbf{A} = 10 * rand(m, n)$, $\mathbf{y} = A * xstar(n, 1)$,
  $\mathbf{xstar}(n, 1) = round(10 * rand(n, 1) + 0.5)$
- $\mathbf{x0}(n, 1) = round(5 * rand(n, 1))$, $\mathbf{r0} = 5 * norm(x0 - xstar)$
- Accuracy $\varepsilon_f = (10^{-6})^p$, $1 \leq p \leq 2$

$n = 20$, $m = 200$, no regularization ($\lambda = 0$)

p	ε_f	*time (sec)*	*itn*	f_p	*dx*
1.0	1.0e-06	1.17	19668	2.3199e-08	5.3e-11
1.2	3.2e-08	1.50	18005	3.7630e-10	4.4e-11
1.5	1.0e-09	1.25	17577	1.4034e-11	6.8e-10
1.8	3.2e-11	1.39	16833	2.6313e-13	9.8e-10
2.0	1.0e-12	0.97	16384	1.0013e-14	2.5e-09

Test example 2: robustness of LMM

- $\mathbf{n} = 20$, $\mathbf{m} = 200$
- $\mathbf{A} = 10 * rand(m, n)$, $\mathbf{y} = A * xstar(n, 1)$, $\mathbf{xstar}(n, 1) = round(10 * rand(n, 1) + 0.5)$
- $\mathbf{x0}(n, 1) = round(5 * rand(n, 1))$, $\mathbf{r0} = 5 * norm(x0 - xstar)$
- even components of $\mathbf{y}$ are multiplied by the value $q = (1.0 + 1.0 * sign(0.5 - rand))$

$n = 20$, $m = 200$, no regularization ($\lambda = 0$)

p	ε_f	$time\ (sec)$	itn	f_r	dx
1.0	1.0e-06	1.13	18890	5.7678e+04	2.2e-10
1.2	3.2e-08	0.93	10978	2.8183e+05	1.2e+01
1.5	1.0e-09	0.88	12441	1.3528e+06	2.8e+01
1.8	3.2e-11	1.21	14321	6.5044e+06	3.4e+01
2.0	1.0e-12	0.97	16383	3.1498e+07	3.6e+01

Content

Test example 3: regularization effect

- $m = 20, \ n = 7$
- Columns $\mathbf{a}^1, \ldots, \mathbf{a}^5$ of the 20×7 matrix $\mathbf{A}$ obtained using $U[0; 50]$
- $\mathbf{y}$ obtained using $\mathbf{a}^1, \ldots, \mathbf{a}^5$, known parameters $(x_1, \ldots, x_5, x_6 = 0, x_7 = 0)$, and $\varepsilon \sim N(0, 100)$

$$y_i = 6.6a_{i1} + 4.4a_{i2} + 2.6a_{i3} + 1.5a_{i4} + 7.3a_{i5} + \varepsilon, \ \ i = \overline{1, 20}$$

- Variant 1: $\mathbf{a}^6 = 0.3\mathbf{a}^1 + 0.2\mathbf{a}^3, \ \mathbf{a}^7 = 0.4\mathbf{a}^4 + 0.2\mathbf{a}^2$
- Variant 2: $\mathbf{a}^6 = 0.8\mathbf{a}^1 + 0.2\mathbf{a}^3, \ \mathbf{a}^7 = 0.4\mathbf{a}^4 + 0.2\mathbf{a}^2$
- $p = 2, \ \varepsilon_f = 10^{-6}$
- $x^* = (6.6, 4.4, 2.6, 1.5, 7.3, 0.0, 0.0)$

Test example 3: regularization effect

	$\lambda = 0$		$\lambda = 0.01$	
	Variant 1	Variant 2	Variant 1	Variant 2
x_n	-2.38e+05	1.73e+05	6.5883	1.0526
	-4.54e+05	6.71e+05	4.4177	4.3894
	-1.58e+05	4.33e+04	2.5880	1.2232
	-9.09e+05	1.34e+06	1.4903	1.5099
	7.91e+00	7.91e+00	7.9229	7.8969
	7.94e+05	-2.16e+05	**0.0000**	6.9286
	2.27e+06	-3.35e+06	**0.0000**	**0.0000**
f_p	2.24e+01	8.75e+00	2.10e+01	1.56e+01
dx	7.0e+06	9.8e+06	2.0e+01	2.0e+01

Conclusions

1. The **emlmpr** algorithm solves linear regression problem with *dozens* of unknown parameters and *hundreds* of observations within 1–2 seconds

2. L_1-regularization is an effective tool for finding parameters of linear regression model if its features are linearly dependent

Acknowledgements

National Research Foundation of Ukraine
(grant №2021.01/0136)

Volkswagen Foundation
(grant №97 775)

Project of Research Works of Young Scientists
(№07-02/03-2023)

DTT TS KNU NASU
(0124U002162, grant №2M-2024)

Questions?

Thank you for your attention

vik.stovba@gmail.com