Using of Ellipsoid Method for Fitting Convex or Concave Quadratic Function

P. Stetsyuk[1] M. Korablov[1]

[1]Department of Nonsmooth Optimization Methods
V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine

9th International Conference on Control and Optimization
with Industrial Applications (COIA 2024)
August 27-29, 2024, Istanbul, Türkiye

Table of Contents

Table of Contents

Introduction

- Regression models are widely used in artificial intelligence, statistical data analysis, finance, economics, medicine and many other areas.
- Most of the times linear models are more than enough to solve this type of supervised learning problem.
- However, in some applications a non-linear dependence between some variables can be clearly seen from the observed data or is a priori known to be present in the data.
- In this work, we study the case when such non-linear dependence is believed to be quadratic.

Table of Contents

Notation

- Let $\left\{\left(x_1^{(k)}, \ldots, x_d^{(k)}, y^{(k)}\right) \in \mathbb{R}^{d+1} : k = \overline{1, m}\right\}$ denote a dataset of size m.
- Here for every measurement $k = \overline{1, m}$ observed values $y^{(k)}$ are dependent on values of d factors $x_1^{(k)}, \ldots, x_d^{(k)}$.
- We have reasons to assume that such dependence is quadratic.

Quadratic function

- General form: $f(x, W, w) = \sum_{i,j=1}^{d} W_{ij} x_i x_j + \sum_{i=1}^{d} w_i x_i + w_0$
  - $x = (x_1, \dots, x_d)^\top \in \mathbb{R}^d$ is a vector of factors;
  - $\{W_{ij}\}_{i,j=1}^{d} \in \mathbb{R}^{d \times d}$, $\{w_i\}_{i=0}^{d} \in \mathbb{R}^{d+1}$ are the unknown coefficients.
- Vector notation: $f(x, W, w) = x^\top W x + w^\top (1 \oplus x)$
  - $1 \oplus x = (x_0 := 1, x_1, \dots, x_d)^\top$;
  - $W = \{W_{ij}\}_{i,j=1}^{d}$ is a symmetric $d \times d$ matrix;
  - $w = \{w_i\}_{i=0}^{d}$ is a $(d+1)$-dimensional vector.

Least moduli criterion powered to p

- A generalized criterion for models parameters estimation, covers both least squares ($p = 2$) and least moduli ($p = 1$), allows $p \in [1; 2]$.

$$F_p(W^*, w^*) = \min_{\substack{W \in \mathbb{R}^{d \times d} \\ w \in \mathbb{R}^{d+1}}} \sum_{k=1}^{m} \left| y^{(k)} - f(x^{(k)}, W, w) \right|^p \qquad (1)$$

- Important aspect of fitting a quadratic model to the data – make sure that some specific relations between factors and observed values are described correctly.

Concavity/convexity constraints

- We can do so by imposing special constraints:
- Constraint for concavity: $C_1(W) = \lambda_{max}(W) \leq \lambda^*$
  - $\lambda_{max}(W)$ is the maximal eigenvalue of W
  - λ^* can take three values to impose different constraints:
    - $\lambda^* = 0$ for concavity;
    - $\lambda^* = -\varepsilon_\lambda$ for strict concavity with ε_λ being a tiny positive number;
    - $\lambda^* = +\infty$ for no concavity restriction.

Concavity/convexity constraints

- We can do so by imposing special constraints:
- Constraint for concavity: $C_1(W) = \lambda_{max}(W) \leq \lambda^*$
  - $\lambda_{max}(W)$ is the maximal eigenvalue of W
  - λ^* can take three values to impose different constraints:
    - $\lambda^* = 0$ for concavity;
    - $\lambda^* = -\varepsilon_\lambda$ for strict concavity with ε_λ being a tiny positive number;
    - $\lambda^* = +\infty$ for no concavity restriction.
- Constraint for convexity: $C_2(W) = \lambda_{min}(W) \geq \lambda_*$
  - $\lambda_{min}(W)$ is the minimal eigenvalue of W
  - λ_* can take three values to impose different constraints:
    - $\lambda_* = 0$ for convexity;
    - $\lambda_* = \varepsilon_\lambda$ for strict convexity with ε_λ being a tiny positive number;
    - $\lambda_* = -\infty$ for no convexity restriction.

Conditional convex optimization problem statements

- For finding concave quadratic function (imposing convex constraint):

$$F_p(W^*, w^*) = \min_{\substack{W \in \mathbb{R}^{d \times d} \\ w \in \mathbb{R}^{d+1}}} F_p(W, w) \tag{2}$$

$$C_1(W) \leq \lambda^* \tag{3}$$

- For finding convex quadratic function (imposing concave constraint):

$$F_p(W^*, w^*) = \min_{\substack{W \in \mathbb{R}^{d \times d} \\ w \in \mathbb{R}^{d+1}}} F_p(W, w) \tag{4}$$

$$C_2(W) \geq \lambda_* \tag{5}$$

Method of nonsmooth penalty functions

- Transform conditional optimization problem (2)-(3) into unconditional problem:

$$P_1(W^*, w^*) = \min_{\substack{W \in \mathbb{R}^{d \times d} \\ w \in \mathbb{R}^{d+1}}} \left\{ \begin{array}{c} F_p(W,w)+ \\ S_1 \cdot \max\{0, C_1(W) - \lambda^*\} \end{array} \right\} \quad (6)$$

- Transform conditional optimization problem (4)-(5) into unconditional problem:

$$P_2(W^*, w^*) = \min_{\substack{W \in \mathbb{R}^{d \times d} \\ w \in \mathbb{R}^{d+1}}} \left\{ \begin{array}{c} F_p(W,w)+ \\ S_2 \cdot \max\{0, -C_2(W) + \lambda_*\} \end{array} \right\} \quad (7)$$

- Here S_1 and S_2 are nonsmooth penalty multipliers chosen to be large enough so that such transformations preserve the set of minimum points of the original conditional problems.

Table of Contents

Shor's ellipsoid method

- Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function and let us denote its minimum value as $f^* = f(x^*)$, where x^* is a minimum point.
- For any $x \in \mathbb{R}^n$, let $g(x)$ denote a subgradient of f at point x, i.e. the following inequality is satisfied: $(x - x^*)^\top g(x) \geq f(x) - f^* \;\; \forall x \in \mathbb{R}^n$.

Shor's ellipsoid method

- Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function and let us denote its minimum value as $f^* = f(x^*)$, where x^* is a minimum point.
- For any $x \in \mathbb{R}^n$, let $g(x)$ denote a subgradient of f at point x, i.e. the following inequality is satisfied: $(x - x^*)^\top g(x) \geq f(x) - f^* \;\; \forall x \in \mathbb{R}^n$.
- The ellipsoid method allows to find an ε-approximation to x^*, i.e. a point x_ε^* s.t. $f(x_\varepsilon^*) - f^* \leq \varepsilon$ for given $\varepsilon > 0$.
- The convergence speed of the algorithm depends on the starting point x_0, a radius r_0, and the approximation accuracy ε.

Shor's ellipsoid method

- Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function and let us denote its minimum value as $f^* = f(x^*)$, where x^* is a minimum point.
- For any $x \in \mathbb{R}^n$, let $g(x)$ denote a subgradient of f at point x, i.e. the following inequality is satisfied: $(x - x^*)^\top g(x) \geq f(x) - f^* \ \forall x \in \mathbb{R}^n$.
- The ellipsoid method allows to find an ε-approximation to x^*, i.e. a point x_ε^* s.t. $f(x_\varepsilon^*) - f^* \leq \varepsilon$ for given $\varepsilon > 0$.
- The convergence speed of the algorithm depends on the starting point x_0, a radius r_0, and the approximation accuracy ε.
- In this work, we propose EMQFLMP algorithm - based on Shor's **e**llipsoid **m**ethod, an algorithm for fitting a **q**uadratic **f**unction using the **l**east **m**oduli powered to **p** criterion. The input parameter of the algorithm is accuracy $\varepsilon_f > 0$ with which $F_p^* = F_p(W^*, w^*)$ must be found.

Shor's ellipsoid method

The Ellipsoid method

Step 0. Choose starting point $x_0 \in \mathbb{R}^n$ and $r_0 > 0$ such that $\|x_0 - x^*\| \le r_0$. Choose $\varepsilon > 0$ and set $B_0 := I_n \in \mathbb{R}^{n\times n}$ (denoting the identity matrix) and $k := 0$.

Step 1. If $\left\|B_k^\top g(x_k)\right\| r_k \le \varepsilon$, then STOP: $k^* := k$, $x_\varepsilon^* := x_k$.

Step 2. Compute next point $x_{k+1} := x_k - h_k B_k \xi_k$, where $\xi_k := \frac{B_k^\top g(x_k)}{\left\|B_k^\top g(x_k)\right\|}$, $h_k := \frac{1}{n+1} r_k$.

Step 3. Update $B_{k+1} := B_k + \left(\sqrt{\frac{n-1}{n+1}} - 1\right)(B_k \xi_k)\, \xi_k^\top$ and $r_{k+1} := \frac{n}{\sqrt{n^2-1}} r_k$.

Step 4. Set $k := k+1$ and go to Step 1.

Table of Contents

Constructing a quadratic objective function of German economic policy in two variables: unemployment rate x_1, in % and inflation rate x_2, in %.

k	$x_1^{(k)}$	$x_2^{(k)}$	$y^{(k)}$	k	$x_1^{(k)}$	$x_2^{(k)}$	$y^{(k)}$
1	6.00	7.00	1	4	8.00	4.00	1
2	2.00	10.00	0	5	10.00	0.00	0
3	4.00	9.00	1	6	6.00	5.00	2

Table 1: Data for estimating German economic policy as a function of 2 variables

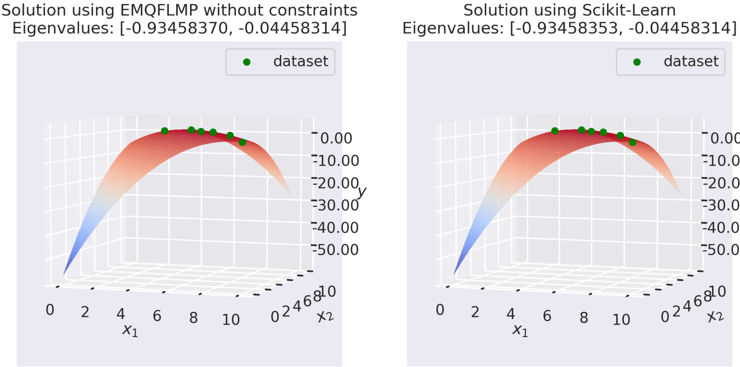

Figure 1: Comparison of two solutions – they turned out to be very close: $\sum_{i,j=1}^{2} |W_{ij}^{(1)} - W_{ij}^{(2)}| + \sum_{i=0}^{2} |w_i^{(1)} - w_i^{(2)}| = 1.66 * 10^{-5}$

Application 1

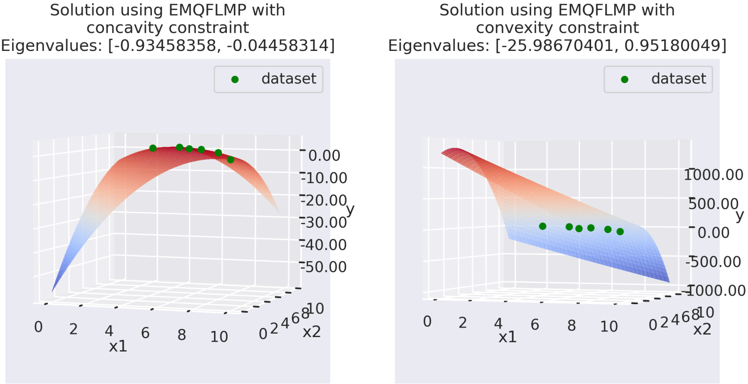

Figure 2: Comparison of two solutions

Application 2

Constructing a quadratic objective function of German economic policy in four variables: inflation rate x_1, in %, unemployment rate x_2, in %, yearly GNP (Gross National Product) growth rate x_3, in % and yearly increase in public debt x_4, in %.

k	$(x_1^{(k)}, \ldots, x_4^{(k)})$	$y^{(k)}$	k	$(x_1^{(k)}, \ldots, x_4^{(k)})$	$y^{(k)}$
1	(4.0, 6.5, 1.0, 4.0)	5.0	9	(5.0, 6.5, 2.0, 4.0)	0.0
2	(7.0, 2.0, 1.0, 4.0)	2.0	10	(2.0, 6.5, 1.0, 7.5)	6.0
3	(6.0, 4.0, 1.0, 4.0)	1.0	11	(7.0, 6.5, 1.0, -5.0)	7.0
4	(3.0, 7.3, 1.0, 4.0)	1.0	12	(5.0, 6.5, 1.0, 1.8)	7.0
5	(-1.0, 9.2, 1.0, 4.0)	6.0	13	(4.0, 10.0, 5.0, 4.0)	2.0
6	(3.0, 5.5, 1.0, 4.0)	10.0	14	(4.0, 10.0, 1.0, -8.0)	0.0
7	(7.0, 6.5, 4.5, 4.0)	3.0	15	(4.0, 6.5, -1.0, -1.0)	0.0
8	(-1.0, 6.5, -2.5, 4.0)	1.0	–	–	–

Table 2: Data for estimating German economic policy as a function of 4 variables

Application 2

- Solution using Scikit-Learn: $F_2(W^*, w^*) = 1.56 * 10^{-23}$

$$W^* = \begin{pmatrix} 8.2835 & 8.4646 & -11.5794 & 2.8131 \\ 8.4646 & 8.2677 & -11.5373 & 2.8763 \\ -11.5794 & -11.5373 & 15.9430 & -5.3432 \\ 2.8131 & 2.8763 & -5.3432 & 1.1549 \end{pmatrix},$$

$$w^* = \begin{pmatrix} 762.5671 \\ -160.7556 \\ -161.5503 \\ 232.5146 \\ -51.9079 \end{pmatrix}$$

- Solution via EMQFLMP without constraints - very similar: $\sum_{i,j=1}^{4} |W_{ij}^{(1)} - W_{ij}^{(2)}| + \sum_{i=0}^{4} |w_i^{(1)} - w_i^{(2)}| = 1.42 * 10^{-5}$

Application 2

$\lambda^* = 0$				
p	$F_p(W^*, w^*)$	$\lambda_{min}(W^*)$	$\lambda_{max}(W^*)$	MAE_{train}
1.0	13,560112	-3.967	-2.47*10^{-16}	0.904007
1.2	17,755620	-4.101	-3.06*10^{-16}	0.938788
1.4	22,166580	-2.074	-8.43*10^{-17}	0.997637
1.6	27,184136	-0.777	-8.92*10^{-18}	1.056110
1.8	33,095451	-0.274	-2.04*10^{-17}	1.111139
2.0	40,182805	-0.295	-7.87*10^{-17}	1.161183

Table 3: Solutions obtained using EMQFLMP with concavity constraint and setting accuracy $\varepsilon_f = 10^{-12}$

Application 2

$\lambda^* = -10^{-9}$				
p	$F_p(W^*, w^*)$	$\lambda_{min}(W^*)$	$\lambda_{max}(W^*)$	MAE_{train}
1.0	13,560112	-3.967	-1.00000077e-09	0.904007
1.2	17,755620	-4.101	-1.00000073e-09	0.938788
1.4	22,166580	-2.074	-1.00000002e-09	0.997637
1.6	27,184136	-0.777	-1.00000012e-09	1.056110
1.8	33,095451	-0.274	-1.00000006e-09	1.111139
2.0	40,182805	-0.295	-1.00000008e-09	1.161183

Table 4: Solutions obtained using EMQFLMP with strict concavity constraint and setting accuracy $\varepsilon_f = 10^{-12}$

Table of Contents

Conclusion

- Special type of regression problem, the task of fitting a concave or convex quadratic function, has been investigated in detail.
- Computationally efficient conditions to ensure concavity/convexity of the fitted function were discussed.
- For solving this problem using the least moduli criterion powered to $p \in [1, 2]$, algorithm based on Shor's ellipsoid method was proposed.
- This algorithm can be used for various machine learning applications since it allows to address many issues of fitting a quadratic function, providing more flexibility and reliability for researchers.

Acknowledgements

DTT TS KNU NASU grant №2M-2024
Volkswagen Foundation grant № 97775

Thank You for Your attention!