# Using the Ellipsoid Method to Find Parameters of Lasso and Ridge Regressions

Petro Stetsyuk, Olha Khomiak

V.M. Glushkov Institute of Cybernetics,
Kyiv, Ukraine

International Conference dedicated to
the 60th anniversary of the foundation of Vladimir Andrunachievici Institute
of Mathematics and Computer Science, MSU (IMCS60)
Chisinau, October 10-13, 2024

# Abstract and keywords

## Abstract

We consider the optimization problem for finding the parameters of a linear regression according to the criterion of the smallest moduli in the degree $p$ $(1 \leq p \leq 2)$ with the regularization of parameters according to the criterion of the smallest modules in the degree $q$ $(1 \leq q \leq 2\ )$. Its partial cases are lasso regression and ridge regression, as well as the method of least squares and the method of least modules. An algorithm for solving the problem was developed based on the well-known ellipsoid method.

## Keywords:

Lasso regression, ridge regression, method of least squares, method of least modules, ellipsoid method

# Content

1. Some facts of the ellipsoid method history
2. The algorithm **emshor** and its **Octave** implementation
3. Using **emshor** for Lasso and Ridge regressions

# Content

# The ellipsoid method was proposed

1976 by **Yudin and Nemirovskii** as a method of successive cutting-plane [*],

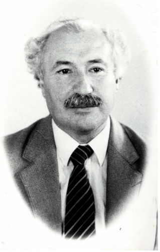
1977 by **Shor** as a variant of the method with space dilation in the direction of the subgradient [**].


* Yudin D.B. and Nemirovskii A.S. *Informational complexity and effective methods for the solution of convex extremal problems* // Ekonom. Mat. Metody, 12, No. 2 (1976).

** Shor N.Z. *Cut-off method with space extension in convex programming problems* // Cybernetics, 13, No. 1 (1977).

# Yudin and Shor „from the banks of the Dnipro“

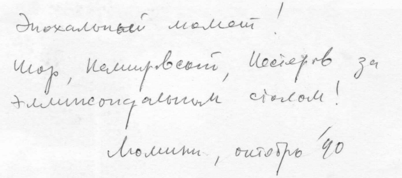
**David Borisovich Yudin**
born May 21, 1919
in Yekaterinoslav (today - Dnipro),
in 1941 graduated from
Dnepropetrovsk University



**Naum Zuselevich Shor**
born January 1, 1937
in Kyiv (city on the Dnipro),
in 1958 graduated from
Kyiv University

# Epochal moment!

N. Shor,
A. Nemirovski,
Y. Nesterov at the
ellipsoidal table!
October 1990


Эпохальный момент!
Шор, Немировский, Нестеров за
эллипсоидальным столом!
Люмини, октябрь '90

# XI ISMP, Bonn, August 23–27, 1982

## Fulkerson Prizes for the ellipsoid method:

**1.** Grötchel M., Lóvasz L., Schrijver A. (**1981**)
**2.** Khachiyan L. (**1979**), Yudin D., Nemirovski A. (**1976**)

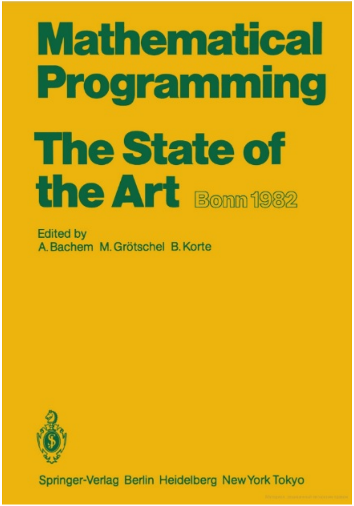
## Shor's plenary report:

„Generalized gradient methods of nondifferentiable optimization employing space dilatation operations", published in [*]

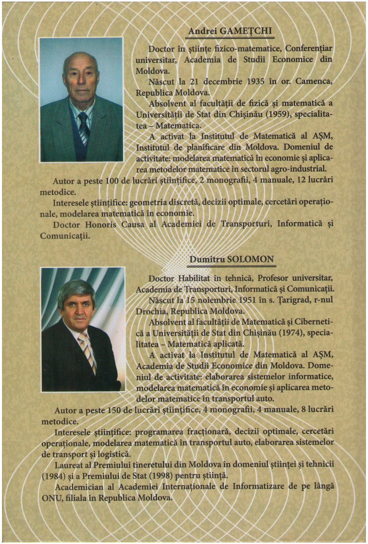
* MATHEMATICAL PROGRAMMING: THE STATE OF ART, BONN, 1982 / *Bachem A., Grötchel M., Korte B. (eds.)* – Berlin: Springer-Verlag, 1983. – 655 p.

# XI International Math. Programming Symposium

N. Shor in Bonn (1982)



# Metoda elipsoidului în Republica Moldova

**Andrei GAMEŢCHI**

Doctor în ştiinţe fizico-matematice, Conferenţiar universitar, Academia de Studii Economice din Moldova.

Născut la 21 decembrie 1935 în or. Camenca, Republica Moldova.

Absolvent al facultăţii de fizică şi matematică a Universităţii de Stat din Chişinău (1959), specialitatea – Matematica.

A activat la Institutul de Matematică al AŞM, Institutul de planificare din Moldova. Domeniul de activitate: modelarea matematică în economie şi aplicarea metodelor matematice în sectorul agro-industrial.

Autor a peste 100 de lucrări ştiinţifice, 2 monografii, 4 manuale, 12 lucrări metodice.

Interesele ştiinţifice: geometria discretă, decizii optimale, cercetări operaţionale, modelarea matematică în economie.

Doctor Honoris Causa al Academiei de Transporturi, Informatică şi Comunicaţii.

**Dumitru SOLOMON**

Doctor Habilitat în tehnică, Profesor universitar, Academia de Transporturi, Informatică şi Comunicaţii.

Născut la 15 noiembrie 1951 în s. Ţarigrad, r-nul Drochia, Republica Moldova.

Absolvent al facultăţii de Matematică şi Cibernetică a Universităţii de Stat din Chişinău (1974), specialitatea – Matematică aplicată.

A activat la Institutul de Matematică al AŞM, Academia de Studii Economice din Moldova. Domeniul de activitate: elaborarea sistemelor informatice, modelarea matematică în economie şi aplicarea metodelor matematice în transportul auto.

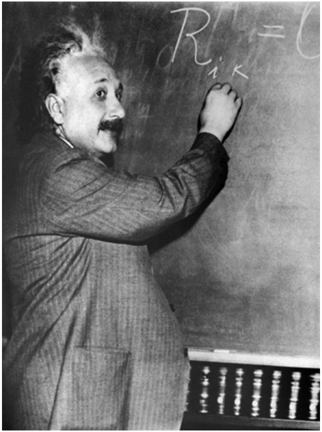
Autor a peste 150 de lucrări ştiinţifice, 4 monografii, 4 manuale, 8 lucrări metodice.

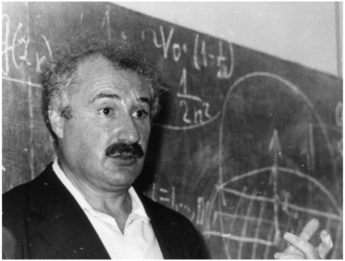
Interesele ştiinţifice: programarea fracţionară, decizii optimale, cercetări operaţionale, modelarea matematică în transportul auto, elaborarea sistemelor de transport şi logistică.

Laureat al Premiului tineretului din Moldova în domeniul ştiinţei şi tehnicii (1984) şi a Premiului de Stat (1998) pentru ştiinţă.

Academician al Academiei Internaţionale de Informatizare de pe lângă ONU, filiala în Republica Moldova.

# Einstein writes a message to Shor

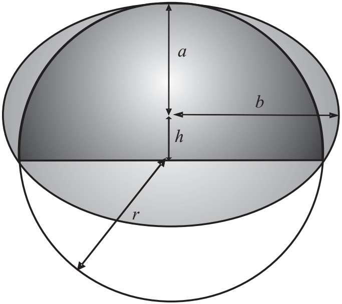




to denote space dilation operator as $R_\alpha(\xi)$

# Content

# The minimum volume ellipsoid (the ellipsoid $\mathcal{E}_n$)




The ellipsoid $\mathcal{E}_n$, containing half-ball $S_n$ in $\mathbb{R}^n$, has following parameters

$$a = \frac{n}{n+1} r, \; b = \frac{n}{\sqrt{n^2-1}} r, \; h = \frac{1}{n+1} r,$$

where $r$ is radius of ball $S_n$.

To transform $\mathcal{E}_n$ into a „new" ball we have to dilate the space with coefficient $\alpha = \frac{b}{a} = \sqrt{\frac{n+1}{n-1}}$, $\alpha > 1$.

## The ratio of $\mathcal{E}_n$ to $S_n$ volumes equals

$$q(n) = \frac{\mathrm{vol}(\mathcal{E}_n)}{\mathrm{vol}(S_n)} = \frac{1}{\alpha}\left(\frac{b}{r}\right)^n = \sqrt{\frac{n-1}{n+1}}\left(\frac{n}{\sqrt{n^2-1}}\right)^n < \exp\left\{-\frac{1}{2n}\right\}$$

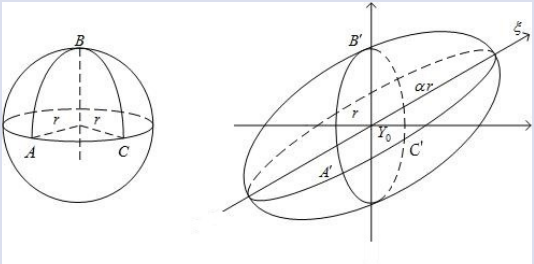
# Shor's space dilation operator

## Space dilation operator has the following form

$$R_\alpha(\xi) = I_n + (\alpha - 1)\xi\xi^T, \quad \text{where} \quad \alpha > 1.$$

Here $\alpha$ is the coefficient of space dilation in the normed direction $\xi \in \mathbb{R}^n$, $\|\xi\| = 1$; $I_n$ is the identity $n{\times}n$-matrix.

## Example in $\mathbb{R}^3$: ball (left) is dilated to ellipsoid (right)



# The algorithm **emshor**

## Convex function minimization problem:

for a convex function $f(x)$, $x \in \mathbb{R}^n$, to find the point $x^*_\varepsilon$, subject to $f(x^*_\varepsilon) - f^* \leq \varepsilon$, where $f^* = f(x^*)$, $\varepsilon > 0$.

## Input parameters:

$x_0$ – starting point, $x_0 \in \mathbb{R}^n$;
$r_0$ – the radius of the ball localizing point $x^* \in \mathbb{R}^n$;
$\varepsilon$ – parameter for stopping: $x^*_\varepsilon$: $f(x^*_\varepsilon) - f^* \leq \varepsilon$.

## Notation:

$g(x_k)$ – subgradient of $f(x)$ at point $x_k$.

# The $B$-form of algorithm **emshor**$(x_0, r_0, \varepsilon)$

**Step 0.** Choose $x_0 \in \mathbb{R}^n$, $r_0 > 0$, $\varepsilon > 0$, $\|x_0 - x^*\| \le r_0$.

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

$$x_{k+1} := x_k - \frac{r_k}{n+1} B_k \xi_k, \quad \text{where} \quad \xi_k := \frac{B_k^\top g(x_k)}{\left\|B_k^\top g(x_k)\right\|}.$$

**Step 3.** Update

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

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

# Octave function emshor

```
# Input parameters:                                                  #row00
# calcfg - name of the function for calculation of f and g           #row00a
# x0 - starting point, x0(1:n)                                       #row00b
# rad - the radius of the ball localizing the minimum point          #row00c
# epsf - parameter for stopping (accuracy)                           #row00d
# maxitn - parameter for stopping (maximal no. of iterations)        #row00e
# intp - printing interval (after each intp iterations)              #row00f
# Output parameters:                                                 #row00g
# x - approximation of the minimum point, x(1:n)                     #row00h
# f - value of function f at the point x                             #row00i
# itn - number of iterations performed                               #row00j
# ist - exit code (1 = epsf, 4 = maxitn)                             #row00k

function[x,f,itn,ist] = emshor(calcfg,x0,rad,epsf,maxitn,intp); #row01
dn=double(length(x0)); beta=sqrt((dn-1.d0)/(dn+1.d0));           #row02
x=x0; radn=rad; B=eye(length(x));                                #row03
for (itn = 0:maxitn)                                             #row04
   [f,g1] = calcfg(x); g=B'*g1; dg=norm(g);                      #row05
   if(radn*dg < epsf) ist = 1;  return; endif                    #row06
   xi=(1.d0/dg)*g; dx = B * xi;                                  #row07
   hs=radn/(dn+1.d0); x -= hs * dx;                              #row08
   B += (beta - 1) * (B * xi) * xi';                             #row09
   radn=radn/sqrt(1.d0-1.d0/dn)/sqrt(1.d0+1.d0/dn);              #row10
   if(mod(itn,intp)==0)                                          #row11
     printf("itn %4d  f %14.6e\n",itn,f);                        #row12
   endif                                                         #row13
endfor                                                           #row14
ist = 4;                                                         #row15
endfunction #emshor                                              #row16
```

# Octave function **emshor** (comments)

```
# Input parameters:
# calcfg - name of the function for calculation of f and g
# x0 - starting point, x0(1:n)
# rad - the radius of the ball localizing the minimum point
# epsf - parameter for stopping (accuracy)
# maxitn - parameter for stopping (maximum of iterations)
# intp - printing interval (after each intp iterations)
# Output parameters:
# x - approximation of the minimum point, x(1:n)
# f - value of function f at the point x
# itn - number of iterations performed
# ist - exit code (1 = epsf, 4 = maxitn)
```

# Octave function **emshor** (program code)

```
function[x,f,itn,ist]
                  = emshor(calcfg,x0,rad,epsf,maxitn,intp);    #row01
dn=double(length(x0)); beta=sqrt((dn-1.d0)/(dn+1.d0));         #row02
x=x0; radn=rad; B=eye(length(x));                              #row03
for (itn = 0:maxitn)                                           #row04
   [f,g1] = calcfg(x); g=B'*g1; dg=norm(g);                    #row05
   if(radn*dg < epsf) ist = 1;  return; endif                  #row06
   xi=(1.d0/dg)*g; dx = B * xi;                                #row07
   hs=radn/(dn+1.d0); x -= hs * dx;                            #row08
   B += (beta - 1) * (B * xi) * xi';                           #row09
   radn=radn/sqrt(1.d0-1.d0/dn)/sqrt(1.d0+1.d0/dn);            #row10
   if(mod(itn,intp)==0)                                        #row11
     printf("itn %4d  f %14.6e\n",itn,f);                      #row12
   endif                                                       #row13
endfor                                                         #row14
ist = 4;                                                       #row15
endfunction #emshor                                            #row16
```

# Content

# Formulation of the optimization problem

Let matrix $A = \{a_{ij}\}_{i=1,\dots,m}^{j=1,\dots,n}$, vector $y = \{y_i\}_{i=1,\dots,m}$, and three scalars $\lambda \geq 0$, $1 \leq p, q \leq 2$ be given.

**Consider the convex optimization problem**

$$f(x_{pq}^*) = \min_{x \in \mathbb{R}^n} \left\{ f(x) = \sum_{i=1}^{m} \left| \sum_{j=1}^{n} a_{ij} x_j - y_i \right|^p + \lambda \sum_{i=1}^{n} |x_i|^q \right\}, \quad (1)$$

where $x = \{x_j\}_{j=1,\dots,n}$ is a vector of unknown parameters.

The convex function $f(x)$ is **smooth** if $p > 1$ and $q > 1$ and **non-smooth** if $p = 1$ or $q = 1$.

# Partial cases of the problem (1)

## Lasso regression: $p = 2$ and $q = 1$

$$f^*_{lasso} = \min_{x \in \mathbb{R}^n} \left\{ f_1(x) = \|y - Ax\|^2 + \lambda \sum_{i=1}^{n} |x_i| \right\}$$

## Ridge regression: $p = 2$ and $q = 2$

$$f^*_{ridge} = \min_{x \in \mathbb{R}^n} \left\{ f_2(x) = \|y - Ax\|^2 + \lambda \sum_{i=1}^{n} x_i^2 \right\}$$

## $\lambda = 0$

least squares method ($p$=2) and least moduli method ($p$=1).

# The formula for calculating $g_f(\overline{x})$ at the point $\overline{x}$

$$g_f(\overline{x}) = p \begin{pmatrix} \sum_{i=1}^{m} \text{sign}\left(\sum_{j=1}^{n} a_{ij}\overline{x}_j - y_i\right)\left|\sum_{j=1}^{n} a_{ij}\overline{x} - y_i\right|^{p-1} a_{i1} \\ \dots \\ \sum_{i=1}^{m} \text{sign}\left(\sum_{j=1}^{n} a_{ij}\overline{x}_j - y_i\right)\left|\sum_{j=1}^{n} a_{ij}\overline{x} - y_i\right|^{p-1} a_{in} \end{pmatrix}$$

$$+\lambda \cdot q \begin{pmatrix} \sum_{i=1}^{n} \text{sign}\left(\overline{x}_1\right)\left|\overline{x}_1\right|^{q-1} \\ \dots \\ \sum_{i=1}^{n} \text{sign}\left(\overline{x}_n\right)\left|\overline{x}_n\right|^{q-1} \end{pmatrix}. \tag{2}$$

# Algorithm for solving the problem (1)

**Step 0.** Choose $x_0 \in \mathbb{R}^n$ and $r_0 > 0$ so that $\left\|x_0 - x^*_{pq}\right\| \le r_0$.
Also, choose $\varepsilon > 0$, set $B_0 := I_n$ and $k := 0$.

**Step 1.** Calculate $g_f(x_k)$ at the point $x_k$ using formula (2).

**Step 2.** If $\left\|B_k^\top g_f(x_k)\right\| r_k \le \varepsilon$, then $k^* := k$, $x^*_\varepsilon := x_k$ and STOP.

**Step 3.** Calculate

$$x_{k+1} := x_k - h_k B_k \xi_k, \quad \text{where} \quad \xi_k := \frac{B_k^\top g_f(x_k)}{\left\|B_k^\top g_f(x_k)\right\|}, \quad h_k := \frac{1}{n+1} r_k.$$

**Step 4.** Update

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

**Step 5.** Set $k := k+1$ and go to Step 1.

## Theorem

*Let $x_k$ and $x_{k+1}$ be generated by the algorithm. Then, the ratio of volumes of the ellipsoids $\mathcal{E}_k = \{x \colon \|B_k^{-1}(x_k - x)\| \le r_k\}$ and $\mathcal{E}_{k+1} = \{x \colon \|B_{k+1}^{-1}(x_{k+1} - x)\| \le r_{k+1}\}$ does not depend on $k$ and is equal to*

$$q_n := \frac{vol(\mathcal{E}_{k+1})}{vol(\mathcal{E}_k)} = \sqrt{\frac{n-1}{n+1}} \left( \frac{n}{\sqrt{n^2-1}} \right)^n < \exp\left\{ -\frac{1}{2n} \right\} < 1.$$

*Moreover, $x^*_{pq} \in \mathcal{E}_k$ holds for all $k = 0, 1, \ldots, k^*$, $x^*_\varepsilon \in \mathcal{E}_{k^*}$ and, if the algorithm stops, then $f(x^*_\varepsilon) \le f^*_{pq} + \varepsilon$ is valid.*

# Test example

**We use the Concrete Compressive Strength dataset**

from the UCI repository (2017). It contains 1030 instances with eight numerical features and one target value.
The target value is the compressive strength of concrete.

**AMD Ryzen 5 4500U 2.38 GHz, 16 GB, GNU Octave 6.2.0**

$n = 8$ and $m = 1030$;
$x_0 = 0$, $r_0 = 100$, $\varepsilon = 10^{-20}$, $itn = 4000 \div 5000$;
Elapsed time is 0.285754 seconds for 5 runnings.

# Computational results

| $(p,q)$ | $(2,1)$ | $(2,2)$ | $(2,1)$ | $(1,1)$ | $(1,1)$ |
|---|---|---|---|---|---|
| $\lambda$ | $10000$ | $10000$ | $0$ | $0$ | $1000$ |
| $x_1^*$ | 1.196e-01 | 1.158e-01 | 1.134e-01 | 1.245e-01 | 1.225e-01 |
| $x_2^*$ | 1.028e-01 | 9.891e-02 | 9.623e-02 | 1.048e-01 | 9.845e-02 |
| $x_3^*$ | 9.222e-02 | 8.444e-02 | 7.932e-02 | 9.242e-02 | 8.468e-02 |
| $x_4^*$ | -1.99e-01 | -1.89e-01 | -1.82e-01 | -2.05e-01 | -1.79e-01 |
| $x_5^*$ | 1.062e-24 | 1.678e-01 | 2.647e-01 | 3.828e-02 | -1.22e-18 |
| $x_6^*$ | 8.360e-03 | 9.587e-03 | 1.029e-02 | 6.364e-03 | 3.927e-03 |
| $x_7^*$ | 1.620e-02 | 1.315e-02 | 1.133e-02 | 1.689e-02 | 1.708e-02 |
| $x_8^*$ | 1.122e-01 | 1.136e-01 | 1.140e-01 | 1.581e-01 | 1.315e-01 |

**Features:**

$x_1$ – cement, $x_2$ – fly ash, $x_3$ – blast furnace slag, $x_4$ – water, $x_5$ – superplasticizer, $x_6$ – coarse aggregate, $x_7$ – fine aggregate, $x_8$ – age

# Conclusion

An algorithm for finding the parameters of lasso and ridge regressions was developed based on **emshor** – Shor's ellipsoid method

# References


- T. Hastie, R. Tibshirani, J. Friedman. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction.* Springer, New York, 2009.
- P. Stetsyuk, V. Stovba, M. Korablov. *Using of Ellipsoid Method for Finding Linear Regression Parameters with L1-regularization* 2nd International Conference on Information Technologies and Their Applications (ITTA 2024), Baku, Azerbaijan, April 23-25, 2024.
- A. Fischer, O. Khomyak, P. Stetsyuk. *The ellipsoid method and computational aspects.* Commun. Optim. Theory, No. 21, 2023, pp. 1–14.

# Thanks

Volkswagen Foundation
(grant No 97 775)

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

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

# Questions?

THANK YOU
FOR YOUR ATTENTION!

e-mail: stetsyukp@gmail.com
khomiak.olha@gmail.com