Using Ellipsoid Method for Nonsmooth Regression Problems*

Petro Stetsyuk[1,*,†], Volodymyr Semenov[2,†] and Viktor Stovba[1,†]

[1] *V.M. Glushkov Institute of Cybernetics of the NASU, Academician Glushkov Avenue, 40, Kyiv, 03187, Ukraine*
[2] *Taras Shevchenko National University of Kyiv, Volodymyrska Street, 60, Kyiv, 01033, Ukraine*

Abstract

Regression type problems are of great interest and importance due to their numerous applications in modern science and technology areas. Both smooth and nonsmooth models of that type require effective methods for finding unknown parameters with sufficient accuracy. Presented in the paper is a general model that covers a few well-known methods, such as LASSO and RIDGE regression. The empq algorithm based on the Shor's ellipsoid method for minimizing the model function is proposed. A brief description of the ellipsoid method is given, as well as convergence theorem, which allows one to estimate computational complexity of the empq algorithm. Results of three computational experiments of using the general model for solving 1D total variation (TV) denoising problem are presented. In the first two experiments piecewise constant and piecewise linear functions are restored, in the third experiment bitcoin open price curve is denoised. Obtained results show prospects for using the general model and the empq algorithm for solving regression and image processing problems.

Keywords

ellipsoid method, nonsmooth regression models, convex optimization, total variation denoising, ROF model, regularization, image processing

1. Introduction

Regression-type problems with nonsmooth functions are known to be one of the main areas of research in mathematical programming and its applications. This is largely motivated by constant emergence of new application areas and computing technologies progress, which provides new computing paradigms – cluster computing architectures, grid- and cloud computing, which correspond well to mathematical ideas underlying non-smooth optimization [1].

Classical regression models, which are an important type of supervised learning problems, have many practical applications. Now, they are one of the most statistically justified, and intensive work on their generalizations and improving accuracy of solutions obtained is actively underway [2]. In particular, one issue of that kind is the study of the family of regression model training methods between the least moduli method and the least squares method. The least moduli method allows one to ensure robust estimation of model parameters, but is more difficult to use than the least squares method due to the nonsmoothness of the loss function.

Using generalized regression models with ability to control several specific parameters permits to have a qualitative impact on solutions changing them depending on need and external conditions. It makes them extremely useful in such areas as digital signal and image processing [3], image compression, compressed sensing etc. Noises of different types appear continuously and even small amounts of them can significantly complicate obtaining accurate solutions. Since these models remain to be convex for all used parameter intervals (but may be nonsmooth though), the development of general and special efficient methods for minimizing such models is of great

**International Workshop on Computational Intelligence, co-located with the IV International Scientific Symposium "Intelligent Solutions" (IntSol-2025), May 01-05, 2025, Kyiv-Uzhhorod, Ukraine*
[1,*] Corresponding author.
stetsyukp@gmail.com (P. Stetsyuk); semenov.volodya@gmail.com (V. Semenov); vik.stovba@gmail.com (V. Stovba)
0000-0003-4036-2543 (P. Stetsyuk); 0000-0002-3280-8245 (V. Semenov); 0000-0003-3023-5815 (V. Stovba)

importance. Moreover, denoising processes for images require significant computational powers because of sizes and detailing of images, so these methods must be able to deal with high-dimensional problems and obtain solutions with sufficient accuracy.

This paper presents a general model that covers some well-known methods like LASSO and RIDGE regression, and proposes a new algorithm based on Shor's ellipsoid method for its minimization. Further, the total variation (TV) denoising problem is considered, as well as the way it can be formulated using the general model proposed. Finally, results of three computational experiments dedicated to 1D TV denoising are presented that show computational possibilities of the algorithm and the model proposed.

2. Formulation of the general model

Let matrices $A=\{a_{ij}\}_{i=\overline{1,m}}^{j=\overline{1,n}}$ and $C=\{c_{ij}\}_{i=\overline{1,k}}^{j=\overline{1,n}}$, vector $y=\{y_i\}_{i=\overline{1,m}}$ be given. We consider the following convex optimization problem:

$$f_{pq}(x)=\sum_{i=1}^{m}\left|y_i-\sum_{j=1}^{n}a_{ij}x_j\right|^p+\lambda\sum_{i=1}^{k}\left|\sum_{j=1}^{n}c_{ij}x_j\right|^q\to\min_{x\in R^n}, \tag{1}$$

where $f_{pq}^*=f(x_{pq}^*)=\min_{x\in R^n}f_{pq}(x)$. Here $x=\{x_i\}_{i=\overline{1,n}}$ is a vector of variables; $\lambda\geq 0$, $1\leq p,q\leq 2$ are given scalars. The function $f_{pq}(x)$ is nonsmooth if $p=1$ and $q=1$, and smooth if $p>1$ and $q>1$. The problem (1) can be rewritten in matrix form as follows:

$$f_{pq}(x)=\|y-Ax\|_p^p+\lambda\|Cx\|_q^q\to\min_{x\in R^n}, \tag{2}$$

where $\|\cdot\|_p^p$ is a p-th power of L_p-norm of a vector $x\in R^n$, which is defined as $\|x\|_p^p=\sum_{i=1}^{n}|x|^p$.

In the problem of finding parameters of regression models $m\times n$-matrix A is usually called a regression or observation matrix, and the first summand of the function $f_{pq}(x)$ is a p-th power of L_p-norm of residuals $y-Ax$ of a regression model. The second summand of the function $f_{pq}(x)$ is considered as regularization addition, where $\lambda\geq 0$ is a regularization parameter. Matrix C of a size $k\times n$ defines interconnections between variables x_j, $j=\overline{1,n}$, of a regularization addition. With some values of parameters p, q set and defined matrix C the model (1) covers some well-known models. In particular, if $C=I$, $p=2$ and $q=1$, we get so called LASSO-regression model [4]:

$$f_{2,1}^{LASSO}(x)=\|y-Ax\|_2^2+\lambda\sum_{i=1}^{n}|x_i|\to\min_{x\in R^n}, \tag{3}$$

which is to minimize nonsmooth function $f_{2,1}^{LASSO}(x)$ and uses L_1-regularization. This method assumes that the coefficients of the model are sparse, meaning that few of them are non-zero. LASSO is closely related to basis pursuit denoising [5] in the field of signal processing and has potential applications in image compression and compressed sensing.

If $C=I$, $p=2$ and $q=2$, we get so called RIDGE-regression model [6]:

$$f_{2,2}^{RIDGE}(x) = \|y - Ax\|_2^2 + \lambda \sum_{i=1}^{n} x_i^2 \rightarrow \min_{x \in R^n}, \tag{4}$$

which is to minimize strictly convex smooth function $f_{2,2}^{RIDGE}(x)$ and uses L_2-regularization. This model is usually used for estimating coefficients of regression model if the independent variables are highly correlated. Also, it can be useful for dealing with multicollinearity problems, which commonly occur in models with large number of parameters [7].

Finally, when $\lambda = 0$, then the problem (1) corresponds to well-known least squares method (LSM) if $p = 2$ and the least moduli method (LMM) if $p = 1$. LSM plays a key role in obtaining estimates and finding unknown parameters in statistics, and LMM has proven to be robust to anomalous observations or outliers [8, 9].

The problem (1) is a problem of unconditional minimization of the convex function $f_{pq}(x)$, subgradient of which at the point $\bar{x}$ is calculated using the following formula:

$$g_{f_{pq}}(\bar{x}_j) = p \sum_{i=1}^{m} \left| \sum_{j=1}^{n} a_{ij} \bar{x}_j - y_i \right|^{p-1} sign\left(\sum_{j=1}^{n} a_{ij} \bar{x}_j - y_i \right) a_{ij} +$$

$$+ \lambda q \sum_{i=1}^{k} \left| \sum_{j=1}^{n} c_{ij} \bar{x}_j \right|^{q-1} sign\left(\sum_{j=1}^{n} c_{ij} \bar{x}_j \right) c_{ij}, \tag{5}$$

or we can write it in matrix-vector form as follows:

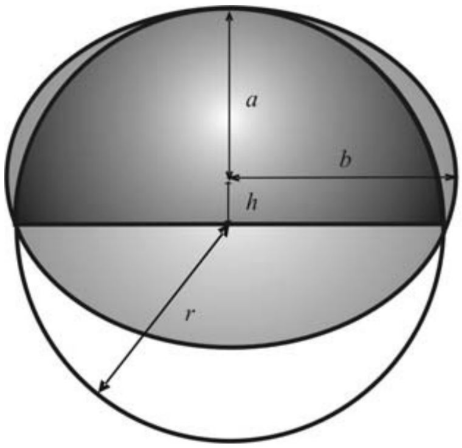
$$g_{f_{pq}}(\bar{x}) = p\, A^T \left(|Ax - y|^{p-1} \odot sign(Ax - y) \right) + \lambda q\, C^T \left(|Cx|^{q-1} \odot sign(Cx) \right), \tag{6}$$

where $|t|$ is the modulus (absolute value) of the number t, and $\odot$ denotes the elementwise Hadamard product.

So, the general model (1) allows us to cover some well-known models and methods with different properties and possibilities. Since the function $f_{pq}(x)$ is convex and can be both smooth or nonsmooth, we can consider using efficient nonsmooth optimization methods for its minimization. In particular, the Shor's ellipsoid method [10, 11, 12] can be used, which is implemented as the emshor program [13]. In the next section a brief description of the ellipsoid methods is given, as well as Octave implementation of the empq algorithm, which is designed for the function $f_{pq}(x)$ minimization.

3. Ellipsoid method and the empq algorithm

Yudin-Nemirovsky-Shor's ellipsoid method is based on using ellipsoid of the minimal volume in E^n, which contains a semi-ball obtained as a result of intersection of n-dimensional ball and half space, which passes through its center. The ellipsoid has a flattened shape in the direction of normal to the hyperplane, which passes through the center of the ball with radius r. Its parameters (see Figure 1) are as follows: a is a length of the minor semi-axis in the direction of normal, which defines a semi-ball; b is a length of the major semi-axis (the number of such axes equals $n-1$); h is a distance from the center of the ball to the center of the ellipsoid in the direction of its minor semi-axis.



$$a = r\frac{n}{n+1}$$

$$b = r\frac{n}{\sqrt{n^2 - 1}}$$

$$h = r\frac{1}{n+1}$$

Figure 1: Ellipsoid of the minimum volume, which contains a semi-ball in E^n.

Iteration of the ellipsoid method is to proceed from the current ellipsoid to the next one with constant coefficient of their volumes decreasing. This coefficient is determined by the ratio of volume of the ellipsoid with semi-axes a and b to the volume of the ball with radius r in E^n and can be written as follows:

$$q_n = \left(\frac{a}{r}\right)\left(\frac{b}{r}\right)^{n-1} = \frac{n}{n+1}\left(\frac{n}{\sqrt{n^2-1}}\right)^{n-1} < 1. \tag{7}$$

It is shown in [13] that

$$q_n < \exp\left\{\frac{-1}{2n}\right\} < 1, \tag{8}$$

so, for the big values of n the coefficient q_n is approximated by the asymptotic formula

$$q_n \approx 1 - \frac{1}{2n}. \tag{9}$$

To use the ellipsoid method for finding the minimum point x^*_{pq} of the problem (1) we must provide it to be localized in n-dimentional ball of radius r_0 with center at the point $x_0 \in R^n$, i.e. $\left\|x_0 - x^*_{pq}\right\| \le r_0$. The algorithm to be used is called the **empq** algorithm, description of which is given below.

The empq algorithm. The input parameter of the algorithm is accuracy $\varepsilon_f > 0$, with which $f^*_{pq} = f_{pq}\left(x^*_{pq}\right)$ is to be found.

Initialization. Let us consider $n \times n$-matrix B and set $B_0 := I_n$, where I_n is $n \times n$ identity matrix. We move to the first iteration with values x_0, r_0 and B_0. Let values $x_k \in R^n$, r_k, B_k be found at the iteration k. Transition to the iteration $k+1$ consists of the following sequence of steps.

Step 1. Calculate $f_{pq}(x_k)$ and subgradient $g_{f_{pq}}(x_k)$ at the point x_k using formula (5) or (6). If $r_k \left\|B_k^T g_{f_{pq}}(x_k)\right\| \le \varepsilon_f$, then “Stop: $k^* = k$ and $x^*_{pq} = x_k$”. Otherwise, proceed to Step 2.

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

Step 3. Calculate the next point

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

Step 4. Calculate

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

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

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

$$\left\| B_k^{-1} (x_k - x_{pq}^*) \right\| \le r_k, \quad k = 0,1,2,\ldots,k^*.$$

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

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

The theorem implies that the ellipsoid method converges with geometric progression rate with coefficient $q_n < \exp\left\{ \frac{-1}{2n} \right\} < 1$ [13]. It also allows us to estimate computational complexity of the algorithm **empq** for finding x_{pq}^* and affirm that it can be successfully run on modern computers, if $n = 30 \div 100$. Indeed, to decrease in 10 times volume of the ellipsoid localizing the point x_{pq}^*, we need to perform K iterations, where $K = \frac{-\ln 10}{\ln q_n} \approx (2 \ln 10) n \approx 4.6 n$. It means that in order to improve deviation of found record value of the function $f_{pq}(x)$ from its optimal value f_{pq}^* by 10 times, it is necessary to perform $4.6 n$ iterations of the algorithm for finding x_{pq}^*.

If $n = 30$ and $\varepsilon_f = 10^{-6} \times f(x_0)$, then the maximal number of iterations of the algorithm is equal to $4.6 n^2 = 46 \times 900 = 41400$. Also, if $n = 100$, the maximal number equals $460\,000$ iterations. Therefore, even the straight-up matrix-vector implementation of calculation of the function $f_{pq}(x)$ value and its subgradient according to the formula (5) allows to provide fast algorithm work on modern computers. Below we will confirm this with the results of computational experiments using Intel Core i7-10750H processor, 2.6 GHz, 16 Gb RAM and GNU Octave 6.3.0 language.

Algorithm for finding an approximation to the point x_{pq}^* is implemented using Octave language. Code of the algorithm is given below.

```
# Input parameters:                                                  #com01
# A(m,n) - observation matrix;                                       #com02
# C(k,n) - regularization summand matrix;                            #com03
# y(m,1) - output vector;                                            #com04
# p,q - scalar parameters, 1<=p<=2, 1<=q<=2;                         #com05
# lambda - regularization rate;                                      #com06
# x0(n,1) - starting point;                                          #com07
# r0 - radius of the ball centered at x0 that localizes x_{pq}^*;  #com08
# epsf, maxitn - stop parameters:                                    #com09
# epsf - precision to stop by the value of the function fpq,         #com10
# maxitn - maximal number of iterations;                             #com11
# intp - print information for every intp iteration.                 #com12
# Output parameters:                                                 #com13
# xpq(n,1) - approximation to x_{pq}^*;                              #com14
# fpq - value of the function f_{pq} at the point xpq;               #com15
# itn - the number of iterations;                                    #com16
# ist - exit code: 1 - epsf, 4 - maxitn.                             #com17
function [xpq,fpq,itn,ist] = empq(y,A,C,p,q,lambda,x0,r0,
                                   epsf,maxitn,intp);                #row01
   n = columns(A); xpq = x0; B = eye(n); r = r0;                     #row02
   dn = double(n); beta = sqrt((dn-1.d0)/(dn+1.d0));                 #row03
   for (itn = 0:maxitn)                                              #row04
     temp1 = A*xpq - y; temp2 = C*xpq;                               #row05
     fpq = sum(abs(temp1).^p) + lambda*sum(abs(temp2).^q);           #row06
     g1 = p*A'*abs(temp1).^(p-1).*sign(temp1) + ...
```

```
               lambda*q*C'*(abs(temp2).^(q-1).*sign(temp2));     #row07
     if((mod(itn,intp)==0)&&(intp<=maxitn))                      #row08
        printf(" itn %4d  fp %14.6e\n",itn,fp);                  #row09
     endif                                                       #row10
     g = B'*g1; dg = norm(g);                                    #row11
     if(r*dg < epsf) ist = 1;  return; endif                     #row12
     xi = (1.d0/dg)*g; dx = B * xi;                              #row13
     hs = r/(dn+1.d0); xpq -= hs * dx;                           #row14
     B += (beta - 1) * B * xi * xi';                             #row15
     r = r/sqrt(1.d0-1.d0/dn)/sqrt(1.d0+1.d0/dn);                #row16
   endfor                                                        #row17
   ist = 4;                                                      #row18
endfunction                                                      #row19
```

Core of the empq program is located in the for loop (rows 4–17). First, the value of the function f (line 6) and its normalized subgradient at the point x_{pq} (row 11) are calculated. If the stop condition is satisfied (row 12), the algorithm stops its work. Stop in the empq algorithm occurs when a condition $r_k \left\| B_k^T g_{f_{pq}}(x_k) \right\| \le \varepsilon_f$ is fulfilled, which is equivalent to condition $f_{pq}(x_k) - f_{pq}^* \le \varepsilon_f$. Otherwise, the next point x_{k+1} is calculated (row 14), the space transformation matrix B_{k+1} (row 15) and the radius r_{k+1} (row 16) are recalculated.

4. Total variation denoising

In addition to machine learning (in particular, regression and classification models, SVM etc.), models of the type (1) are commonly used in various applied fields, such as *signal processing*. This domain belongs to an electrical engineering area and focuses on processing, analyzing and synthesizing signals of different nature, such as scientific measurements, sounds, images, currency fluctuations [15] etc.

The problem of interest is total variation (TV) denoising (filtering), which is a noise removal process. Its main principle claims that signals with excessive and probably incorrect details have rather high total variation. The problem considered is to reduce the variation (and therefore unnecessary details) subject to staying as close as possible to the original signal and preserving important details, such as edges. This allows us to achieve, in particular, digital storage efficiency, since signal preservation in such form requires much less memory, and analysis simplicity. The concept described is known today as ROF model [16].

The total variation problem can be formulated using the described model of the problem (1). We need to find approximation x_i, $i=\overline{1,n}$, to known signal y_i, $i=\overline{1,n}$, and the first summand of the function $f_{pq}(x)$ denotes the closeness measure of them. The second summand represents the total variation and can be defined in different ways using matrix C form. So, the total variation problem could be rewritten using model (1) in the following way:

$$f_{pq}(x) = \sum_{i=1}^{n} |y_i - x_i|^p + \lambda \sum_{i=1}^{k} \left| \sum_{j=1}^{n} c_{ij} x_j \right|^q \to \min_{x \in R^n}, \qquad (10)$$

Here matrix A is of size $n \times n$ and equals to the identity matrix, $\lambda \ge 0$ is a regularization parameter, which is used to regulate intensity of denoising. For instance, if $p=2$, $q=1$, and we use the following $(n-1) \times n$-matrix C_1:

$$C_1 = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 1 & -1 \end{pmatrix}$$

which is two-diagonal with band $(1,-1)$ (all the other elements of which equal zero), model $f_{pq}(x)$ turns into the following model:

$$f_{2,1}(x;C_1)=\sum_{i=1}^{n}(y_i-x_i)^2+\lambda\sum_{i=1}^{n-1}|x_i-x_{i+1}|. \quad (11)$$

As a result of $C_1 x$ multiplication, we get vector $C_1 x=(x_1-x_2, x_2-x_3,\ldots,x_{n-1}-x_n)^T$, which consists of pairwise differences between consecutive elements of vector x_i, $i=\overline{1,n}$. Thus, the second summand in (11) represents the total variation, and since the function $f_{2,1}(x;C_1)$ is to be minimized, the vector $C_1 x$ is expected to be sparse if the parameter λ is big enough. So, we can regulate denoising level using parameter λ: if $\lambda=0$, there is no smoothing and we just restore the signal y_i, $i=\overline{1,n}$, precisely. Otherwise, as $\lambda\to\infty$, the total variation decreases, and filtering process works more intensive. As a result of this process, we get piecewise constant function.

Another form of matrix C that can be used is as follows:

$$C_2=\begin{pmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & 0 & 1 & -2 & 1 \end{pmatrix}.$$

Matrix C_2 is tridiagonal matrix of size $(n-2)\times n$ with band $(1,-2,1)$. If $p=2$ and $q=1$, we get the following model:

$$f_{2,1}(x;C_2)=\sum_{i=1}^{n}(y_i-x_i)^2+\lambda\sum_{i=1}^{n-2}|x_i-2x_{i+1}+x_{i+2}|. \quad (12)$$

In this case, elements of sparse vector $C_2 x$ can be interpreted as difference analogues of the second derivative, and we get piecewise-linear function as a result of filtering process. Moreover, if $q=2$, we talk about smooth second derivative approximation, which is used to obtain smoother piecewise function as a result.

Total variation denoising technique is of particular interest in the field of image processing, where noisy (stained) or corrupted images are to be cleaned and restored. TV-regularization allows to smooth away noise and preserve edges, unlike famous linear smoothing or median filtering techniques [17]. In this case, a 2D signal $y_{i,j}$ is considered, and the total variation, according to [16], could be written in the following form:

$$V(y)=\sum_{i,j}\sqrt{|y_{i+1,j}-y_{i,j}|^2+|y_{i,j+1}-y_{i,j}|^2}, \quad (13)$$

or as its anisotropic version

$$V_{an}(y)=\sum_{i,j}|y_{i+1,j}-y_{i,j}|+|y_{i,j+1}-y_{i,j}|. \quad (14)$$

Such a problem still can be formulated using model (1) with $p=2$ and $q=1$. Common methods used for solving this problem are primal-dual interior-point methods [18] or the split-Bregman method [19]. It also should be noted that there is a popular approach based on reducing nonsmooth problems (11) and (12) to saddle form. For instance, we can rewrite (11) as

$$\min_{x\in R^n}\max_{p\in B_\infty}\left(\|x-y\|_2^2+(C_1 x, p)\right), \quad (15)$$

where $B_\infty=\left\{p\in R^n : \max_{1\le i\le n}|p_i|\le\lambda\right\}$. Then, to obtain approximate solution of the smooth minmax problem one could use extragradient type methods [20, 21] or proximal primal-dual algorithms [22].

Since the problems considered remain convex (but not necessarily smooth) if $p \geq 1$ and $q \geq 1$, some efficient nonsmooth optimization methods can be used for their solving, such as the ellipsoid method (see Section 3) or Shor's r-algorithms [12].

5. Computational experiments

In this section, we present the results of computational experiments obtained using the **empq** algorithm for 1D TV denoising problem.

Test 1. Let us consider piecewise constant function $f_1(x)$, domain of which is divided into three intervals $[0,40]$, $[40,70]$, and $[70,100]$. For each of the intervals j, $j=\overline{1,3}$, value of the function $f_1(x)$ is calculated via the formula

$$f_1^j(x) = randint_j(30) + sign(u)(|u|+3),$$

where $randint_j(30)$ denotes a random integer from the interval $[1,30]$, and u is a random number from the uniform distribution on the interval $[-1,1]$; $sign$ and $|\cdot|$ are the signum function and the absolute value function respectively. Such construction of the range of the function $f_1(x)$ means that to each of the constant functions of the intervals j, $j=\overline{1,3}$, a random noise is added. To restore the original piecewise constant function $f_1(x)$, model (11) is used, which corresponds to the model (1) with $p=2$, $q=1$ and $C=C_1$. Results of the **empq** algorithm work for this problem with two values of λ are presented in Figure 2.

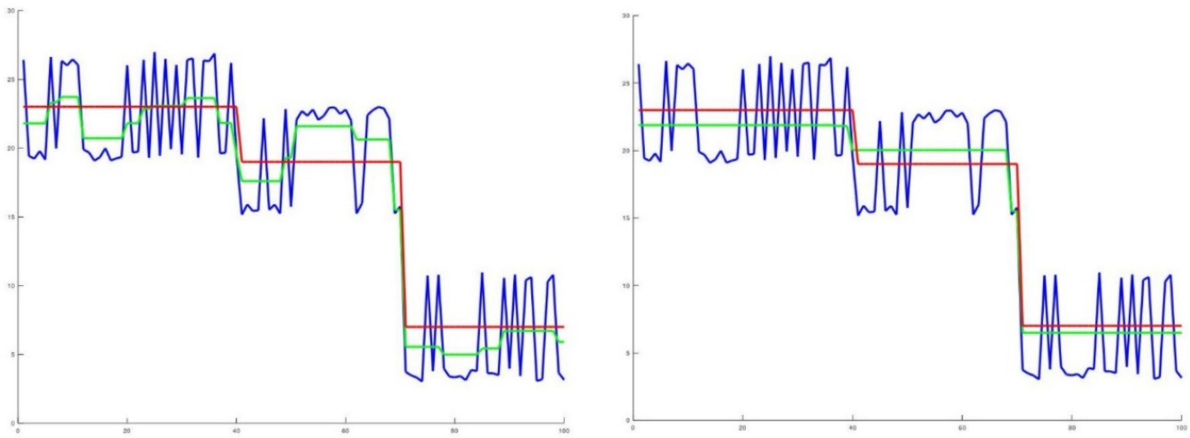


Figure 2: Results of TV denoising of the piecewise constant function $f_1(x)$ using the empq algorithm and model (11) with $\lambda=10$ (left) and $\lambda=50$ (right) ($n=100$).

In Figure 2, the initial piecewise constant function is colored red, its noised version $f_1(x)$ is colored dark blue, and its denoised version, obtained using the **empq** algorithm, is colored green. As can be seen, with $\lambda=10$ we get the restored curve being close to the initial noisy curve; it strives to duplicate ups and downs of the second showing its trend. On the contrary, using parameter value $\lambda=50$ allows one to restore the initial piecewise constant function; selection of λ makes it possible to regulate the intensity of noise filtering.

Test 2. Consider the following piecewise linear function $f_2(x)$:

$$f_2(x) = \begin{cases} x, if\ x \in [1,30] \\ -3x+120, if\ x \in [31,60] \\ -60, if\ x \in [61,80] \\ 2x-220, if\ x \in [81,100] \end{cases}.$$

To simulate noise, we use the same value $sign(u)(|u|+4)$ from the previous test, where u is a random number from the uniform distribution on the interval $[-1,1]$. We use model (12), i.e. parameters $p=2$, $q=1$ and $C=C_2$. Results of filtering of noisy piecewise linear function using the **empq** algorithm for two values of λ are presented in Figure 3. Here, the initial piecewise linear function is colored red, its noisy version $f_2(x)$ is colored blue, and the denoised version of the initial function is colored green.

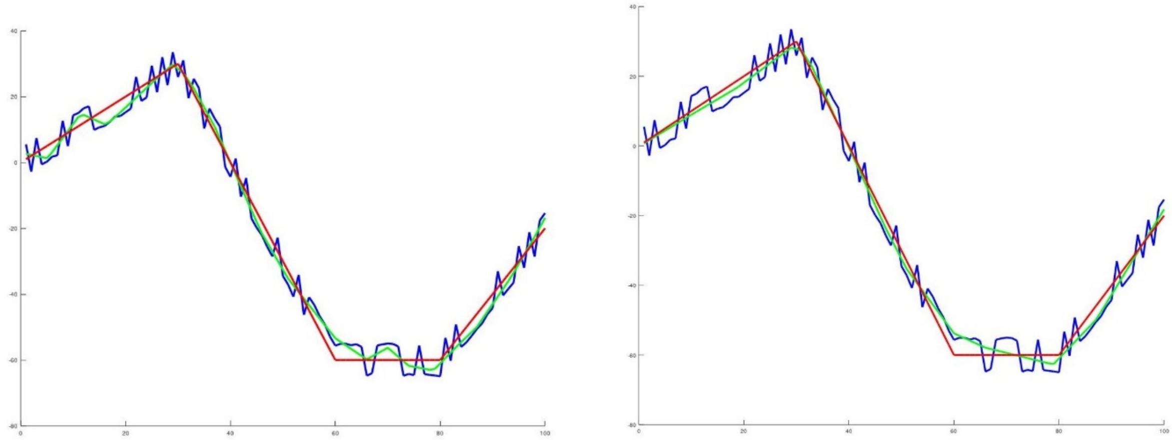


Figure 3: Results of TV denoising of the piecewise linear function $f_2(x)$ using the empq algorithm and model (12) with $\lambda=10$ (left) and $\lambda=50$ (right) ($n=100$).

As can be seen from Figure 3, using of matrix C_2 works better, if a piecewise linear function is denoised rather than a piecewise constant one. With $\lambda=10$ we observe how the denoised curve tends to duplicate the noised one (left picture in Figure 3), but with $\lambda=50$ we get denoised piecewise linear function being much closer to the initial one (right picture in Figure 3). It should be also noted that choosing parameter $q=2$ in model (1) means using smooth 2-derivative approximation, so the denoised curve obtained has edges smoothed away. So, using parameters p and q, as well as regularization coefficient λ and matrix C allows one to control the level of denoising process.

Test 3. Along with a bit synthetical examples, which are though quite important for demonstration of structure of the models described, let us consider results of computational experiments with real datasets, namely bitcoin price evolution. The dataset taken has been collected from the Binance API with open price data captured at one-minute intervals from 11:40 to 13:19 on 01/08/2023. We use model (1) with $p,q=1,2$, $\lambda\in\{10,30,50,70\}$, and matrix $C=\{C_1,C_2\}$. The results of using the **empq** algorithm for given bitcoin dataset denoising is presented in Figure 4 and Figure 5. Here, bitcoin open price data is colored blue, and denoised curve is colored green.

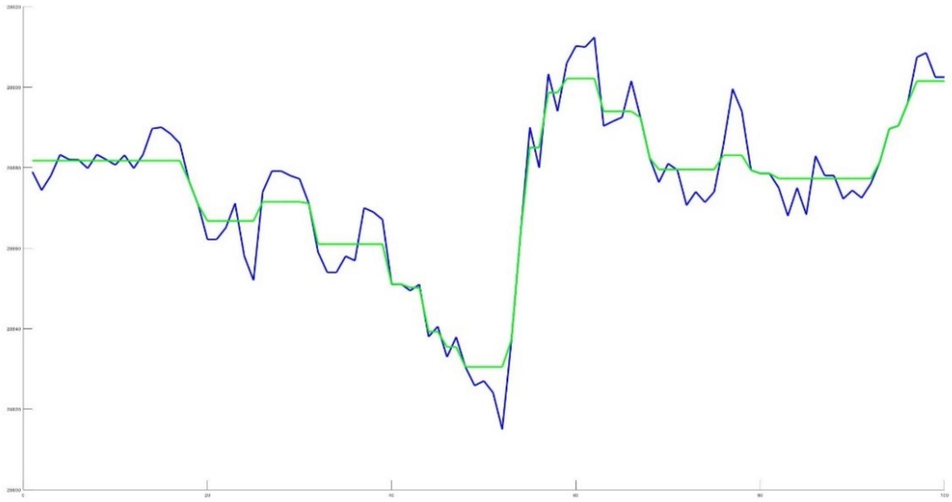


Figure 4: Result of TV denoising of bitcoin open price dataset using the empq algorithm and model (1) with $p=2$, $q=1$, $\lambda=30$, and $C=C_1$ ($n=100$).

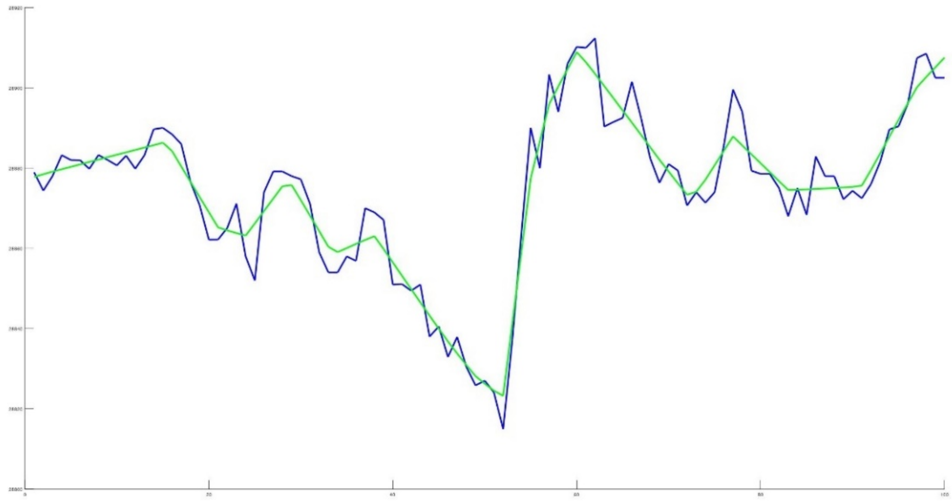


Figure 5: Result of TV denoising of bitcoin open price dataset using the empq algorithm and model (1) with $p=2$, $q=1$, $\lambda=30$, and $C=C_2$ ($n=100$).

As mentioned before, using matrix $C=C_1$ we obtain piecewise constant function as a result of denoising process (see Figure 4), and using matrix $C=C_2$ we get piecewise linear function approximation (see Figure 5), which fits the data much better. In the first case, further increase of λ leads to ignoring important details and edges in data, since the resulting curve tends to be just constant. In the second case, this tendence remains, but we get more details preserved in the ups and downs of a piecewise linear function. Finally, using smooth 2-derivative with $p=2$, $q=2$ and matrix $C=C_2$ gives us smoothed curve as a result (see Figure 6).

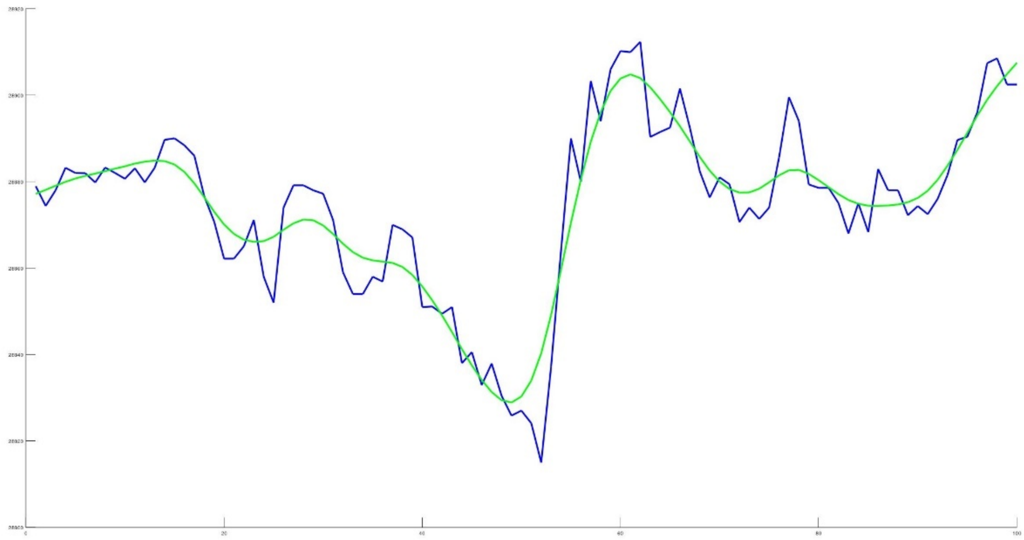


Figure 6: Result of TV denoising of bitcoin open price dataset using the empq algorithm and model (1) with $p=2$, $q=2$, $\lambda=30$, and $C=C_2$ ($n=100$).

Conclusion

The use of different smooth and nonsmooth regression models is rather common not only in modern areas of artificial intelligence and machine learning, but also in a number of applied fields, such as image processing and compression, compressed sensing, noise reduction etc. Problems that appear in these fields are often formulated as nonsmooth or convex optimization problems, so corresponding methods of that kind can be applied.

Proposed **empq** algorithm, which is based on the ellipsoid method, is capable of solving convex optimization problems corresponding to total variation denoising problems if the number of points in signal sample is not greater than $n=100 \div 120$. This number is possible to be increased to some level via using more advanced computing capabilities. The general model considered allows one to regulate denoising level and smoothness of a resulting curve by changing parameters p, q, coefficient λ and choosing the proper form of matrix C to determine interconnections between elements of variables vector in regularization summand. Results of computational experiments demonstrate the prospect and flexibility of such models.

As mentioned before, for solving the problems considered it is appropriate to use other types of methods, such as splitting methods [23], various extragradient schemes, Shor's r-algorithms [11, 12] etc. The last ones are based on using space transformation procedure in the direction of two successive subgradients and provide accelerated convergence while minimizing nonsmooth convex functions of thousands of variables. But the main advantage of the algorithm proposed is obtaining a solution with any given accuracy, whereas other methods are usually limited in this aspect. Nevertheless, the applications of these methods to the problem considered are of a great interest; investigation and comparison of methods efficiency will be conducted in further publications.

Acknowledgements

The paper is supported by the NASU grant for research laboratories/groups of young scientists №02/01-2024(5), and the DTT TS KNU NASU project № 0124U002162.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] I.V. Sergienko, Methods of Optimization and Systems Analysis for Problems of Transcomputational Complexity, Springer, New York, NY, USA, 2012.

[2] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed., Springer Series in Statistics, Textbook, Springer, Berlin/Heidelberg, Germany, 2016.

[3] R. Gonzalez, Digital image processing, New York, NY, Pearson, 2018.

[4] Tibshirani, Robert, Regression Shrinkage and Selection via the lasso, Journal of the Royal Statistical Society, Series B (methodological) 58 (1) (1996) 267–288. doi:10.1111/j.2517-6161.1996.tb02080.x. JSTOR 2346178.

[5] Chen, Shaobing, Donoho, D. Basis pursuit, in: Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers, Vol. 1, 1994, pp. 41–44. doi:10.1109/ACSSC.1994.471413.

[6] A. E. Hoerl, R. W. Kennard, Ridge regression: Biased estimation for nonorthogonal problems, Technometrics 12 (1) (1970) 55–67. doi:10.1080/00401706.1970.10488634.

[7] P. Kennedy, A Guide to Econometrics, Fifth ed., Cambridge, The MIT Press, 2003.

[8] P.J. Huber, E.M. Ronchetti, Robust Statistics, John Wiley & Sons, 2nd Edition, 2011.

[9] J. Fan, P. Hall, On curve estimation by minimizing mean absolute deviation and its implications, The Annals of Statistics (1994) 867–885.

[10] N.Z. Shor, Cutting-off Method with Space Dilation for Solving Convex Programming Problems, Cybernetics (1977) 94–95.

[11] N.Z. Shor, Nondifferentiable Optimization and Polynomial Problems, Kluwer, Amsterdam, 1998.

[12] N.Z. Shor, Minimization Methods for Non-Differentiable Functions, Berlin, SpringerVerlag, 1985.

[13] Martin Grötschel, László Lovász, Alexander Schrijver, Geometric Algorithms and Combinatorial Optimization, Second Edition, Springer-Verlag, Berlin, Heidelberg, 1993.

[14] P. Stetsyuk, A. Fischer, O. Khomyak, The Generalized Ellipsoid Method and Its Implementation, Communications in Computer and Information Science (2020) 355–370. doi:10.1007/978-3-030-38603-0_26.

[15] Alan V. Oppenheim, Ronald W. Schafer, Discrete-Time Signal Processing, Prentice Hall, 1989.

[16] L. I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D. 60 (1–4) (1992) 259–268.

[17] D. Strong, T. Chan, Edge-preserving and scale-dependent properties of total variation regularization, Inverse Problems 19 (6) (2003) S165 – S187.

[18] S. Mehrotra, On the Implementation of a Primal-Dual Interior Point Method, SIAM Journal on Optimization 2 (4) (1992) 575–601.

[19] T. Goldstein, S. Osher, The Split Bregman Method for L1-Regularized Problems, SIAM J. Imaging Sci. 2 (2) (2008) 323–343.

[20] Yu. Malitsky, Projected Reflected Gradient Methods for Monotone Variational Inequalities, SIAM Journal on Optimization 25 (2015) 502–520.

[21] S. V. Denisov, D. A. Nomirovskii, B. V. Rublyov, V. V. Semenov, Convergence of Extragradient Algorithm with Monotone Step Size Strategy for Variational Inequalities and Operator Equations, Journal of Automation and Information Sciences 51 (2019) 12–24. doi:10.1615/JAutomatInfScien.v51.i6.20.

[22] A. Chambolle and T. Pock, A first-order primal-dual algorithm for convex problems with applications to imaging, J. Math. Imaging. Vis. 40(1) (2011) 120–145.

[23] G. Li, H. Wu, Splitting Methods for Nonconvex and Nonsmooth Optimization, Encyclopedia of Optimization (2022) 1–19. doi: 10.1007/978-3-030-54621-2_847-1.