www.czasopisma.pan.pl PAN POLSKA AKADEMIA NAUK www.journals.pan.pl

BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 60, No. 3, 2012
DOI: 10.2478/v10175-012-0061-2

Generalized ordered linear regression with regularization

J.M. ŁĘSKI[1]* and N. HENZEL[2]

[1] Institute of Electronics, Silesian University of Technology, 16 Akademicka St., 44-100 Gliwice, Poland
[2] Institute of Medical Technology and Equipment, 118A Roosevelt St., 41-800 Zabrze, Poland

Abstract. Linear regression analysis has become a fundamental tool in experimental sciences. We propose a new method for parameter estimation in linear models. The 'Generalized Ordered Linear Regression with Regularization' (GOLRR) uses various loss functions (including the ϵ-insensitive ones), ordered weighted averaging of the residuals, and regularization. The algorithm consists in solving a sequence of weighted quadratic minimization problems where the weights used for the next iteration depend not only on the values but also on the order of the model residuals obtained for the current iteration. Such regression problem may be transformed into the iterative reweighted least squares scenario. The conjugate gradient algorithm is used to minimize the proposed criterion function. Finally, numerical examples are given to demonstrate the validity of the method proposed.

Key words: linear regression, IRLS, OWA, conjugate gradient optimization, robust methods.

1. Introduction

From the times of A. Legendre [1] and K. Gauss [2] the least sum of squares (least squares in short) method has been an essential tool in the experimental sciences such as astronomy, biology, physics, sociology, psychology, naming just a few of them. The method had also a major impact on the development of the estimation theory [3]. Currently, the method is routinely applied in signal processing, pattern recognition, machine learning, system identification, fuzzy systems, neural networks and elsewhere where we analyze distorted data [4–9]. Despite the fact that more than 200 years have passed, the least squares criterion is still the most widely used one due to its elegant analytical solution that requires low computational effort. However, it is well known that this method is optimal only for normal (Gaussian) density function of model residuals. In general, for nongaussian noise the maximum likelihood estimator is optimal when the squared residuals are replaced by another function of the residuals, that is, the negative logarithm of the probability density function of the residuals. The Laplacian density function, which is more heavy-tailed than the Gaussian density function, leads to the very popular Least sum of Absolute Deviation method (LAD).

Usually, the density function of the residuals is unknown. To overcome this problem, we need the so-called robust estimator. According to Huber [10, 11], a robust method should have the following properties: (i) it should have a reasonably good accuracy at the assumed model, (ii) small deviations from the model assumptions should impair the performance only by a small amount, (iii) larger deviations from the model assumptions should not cause a catastrophe. Huber proposes to approximate the unknown density function by a linear combination of a certain fixed density and an arbitrary density. If we choose the normal density for a fixed density, then the Huber loss function is obtained (smooth combination of quadratic and linear functions).

Another disadvantage of the least square estimator is its sensitivity to the presence of outliers, that is, atypical, impossibly large for a model, erroneous observations. Up to now, a lot of methods immune to outliers have been proposed. An overview of these methods can be found in [12]. Among these methods, the Least Median of Squares (LMS), the Least Trimmed Squares (LTS) are the most successful ones. These methods fit the model to the majority of the data, by working on the ordered squared residuals. Outliers may be detected as points that lie far away from the robust fit [12]. For the last few years, there has been an increasing interest in incorporating the main result of the statistical learning theory, i.e. the fact that the generalization ability of a model depends both on the empirical risk on a training set and on the complexity of this model [7, 13], to the pattern recognition and regression analysis. The support vector regression uses an ϵ-insensitive loss function and ℓ_2 regularization to control the complexity of a model. The idea of tolerant learning has also been used to introduce the ϵ-insensitive fuzzy modeling [4, 5].

The traditional least sum of squares criterion may also be viewed as a scalar quality of the fit, with the arithmetic mean (after dividing by the number of the data samples) used to aggregate the fit measure for all data samples. If the generalized mean is used as an aggregation operator, then a wide class of the mean operators is embraced, including: arithmetic, harmonic, geometric and more generally root-power mean. In 1988 R. Yager proposed Ordered Weighted Averaging (OWA)[14]. In this case, the importance of the aggregated sample depends on its position after the ordering operation. This class of operators includes for example: min, max and median. An overview of the aggregation operators can be

*e-mail: jleski@polsl.pl

found in [15]. In 2005 OWA was used as a robust estimator of a location parameter to determine the baseline drift of biomedical signal in a moving window [16]. More recently, Yager has proposed OWA-based regression using the power function of the residuals [17].

The use of Iteratively Reweighted Least Squares (IRLS) traces back several decades. In 1973, E.J. Schlossmacher [18] proposed the use of the IRLS technique in the least sum of absolute deviation method of curve fitting. In [19] and [10] this technique was used for robust regression.

The main goal of this work is to show that the regression problems with various loss functions (including the ϵ-insensitive ones), ordered weighted averaging of the residuals, and regularization may be transformed into the iterative reweighted least squares scenario. The conjugate gradient algorithm is used to minimize the proposed criterion function. The second goal of this work is to investigate the performance of the proposed regression method when applied to data in the presence of noise and outliers. Some ideas from this paper have been used in the previous work [6] to design a binary classifier and here are extended to the ordered linear regression.

The remainder of this paper is organized as follows: Sec. 2 presents application of an iteratively reweighted least square criterion function with regularization to estimation of a linear model. Section 3 shows that this approach may be extended to ϵ-insensitive loss functions. Section 4 presents the use of the conjugate gradients for minimization of the criterion function. Section 5 presents simulation results and a discussion on the estimation of a linear models for synthetic datasets in the presence of outliers. Finally, conclusions are drawn in Sec. 6.

2. Problem formulation and initial considerations

We consider the classical regression situation: we have the data (the training set) $\mathcal{T}^{(N)} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)\}$, where N stands for data cardinality, and each independent input datum $\mathbf{x}_i \in \mathbb{R}^t$ (regressors) has a corresponding dependent output datum $y_i \in \mathbb{R}$. From a set of linear models $y = \mathbf{w}^\top \mathbf{x} + w_0$ parameterized by a vector $\mathbf{w}$ and a scalar (bias) w_0, we seek a vector $\mathbf{w}^\star$ and a scalar $w_0^\star$ such that corresponding model fits the data $\mathcal{T}^{(N)}$ best. In the Ordinary Least Squares (OLS) method the quality of the fit is measured by the residual squared error. Defining the augmented data vector $\widetilde{\mathbf{x}}_i = \left[\mathbf{x}_i^\top, 1\right]^\top$ and the augmented vector of parameters $\widetilde{\mathbf{w}} = \left[\mathbf{w}^\top, w_0\right]^\top \in \mathbb{R}^{t+1}$, the linear model can be written as $y = \widetilde{\mathbf{w}}^\top \widetilde{\mathbf{x}}$. The residual for ith data pair equals $e_i = \widetilde{\mathbf{w}}^\top \widetilde{\mathbf{x}}_i - y_i$. In the OLS we seek a vector of parameters $\widetilde{\mathbf{w}}$, by minimizing

$$J\left(\widetilde{\mathbf{w}}\right) = \sum_{i=1}^{N} \left(\widetilde{\mathbf{w}}^\top \widetilde{\mathbf{x}}_i - y_i\right)^2, \tag{1}$$

Let $\mathbf{X}$ be the $N \times (t+1)$ matrix

$$\mathbf{X}^\top \triangleq \left[\widetilde{\mathbf{x}}_1, \widetilde{\mathbf{x}}_2, \cdots, \widetilde{\mathbf{x}}_N\right]. \tag{2}$$

and $\mathbf{y}$ be N-dimensional vector $\mathbf{y}^\top = [y_1, y_2, \cdots, y_N]$. Now, criterion function (1) can be written as

$$J\left(\widetilde{\mathbf{w}}\right) = \left(\mathbf{X}\widetilde{\mathbf{w}} - \mathbf{y}\right)^\top \left(\mathbf{X}\widetilde{\mathbf{w}} - \mathbf{y}\right). \tag{3}$$

The traditional method to prevent over-fitting and make a solution more stable for ill-conditioned (numerically unstable) problems is Tikhonov (ℓ_2) regularization [7]. From the statistical learning theory perspective, regularization is used to improve generalization ability of a model [13]. Criterion function (3) with regularization takes the form

$$J\left(\widetilde{\mathbf{w}}\right) = \frac{1}{N}\left(\mathbf{X}\widetilde{\mathbf{w}} - \mathbf{y}\right)^\top \left(\mathbf{X}\widetilde{\mathbf{w}} - \mathbf{y}\right) + \tau \mathbf{w}^\top \mathbf{w}. \tag{4}$$

Parameter $\tau \geq 0$ controls the trade-off between the complexity of a model and the amount up to which errors are tolerated. Factor $1/N$ is used to make the value of the regularization parameter independent from the cardinality of the dataset.

In the proposed method of linear regression, various loss functions and the idea of regularization are used. We seek vector $\widetilde{\mathbf{w}}$ by the following minimization

$$\min_{\widetilde{\mathbf{w}} \in \mathbb{R}^{t+1}} J\left(\widetilde{\mathbf{w}}\right) \triangleq \frac{1}{N} \sum_{i=1}^{N} h_i \, \mathcal{L}\left(\widetilde{\mathbf{w}}^\top \widetilde{\mathbf{x}}_i - y_i\right) + \tau \, \mathbf{w}^\top \mathbf{w}, \tag{5}$$

where $\mathcal{L}(\cdot)$ stands for a loss function used, and h_i is a weight corresponding to the ith datum (its role is explained later). If we choose the quadratic loss function then in matrix notation (5) takes the form

$$\min_{\widetilde{\mathbf{w}} \in \mathbb{R}^{t+1}} J\left(\widetilde{\mathbf{w}}\right) \triangleq \frac{1}{N}\left(\mathbf{X}\widetilde{\mathbf{w}} - \mathbf{y}\right)^\top \mathbf{H}\left(\mathbf{X}\widetilde{\mathbf{w}} - \mathbf{y}\right) + \tau \widetilde{\mathbf{w}}^\top \widetilde{\mathbf{I}} \widetilde{\mathbf{w}}, \tag{6}$$

where $\mathbf{H} = \operatorname{diag}\left(h_1, h_2, \cdots, h_N\right)$ and $\widetilde{\mathbf{I}}$ is the identity matrix with the last element on the main diagonal set to zero (to make a bias term unregularized). The role of h_is parameters may be threefold: (i) they may correspond to our confidence to the ith datum (${}^c h_i \in [0,1]$), (ii) through the proper selection of the parameters values we may change different loss functions to the quadratic loss (${}^l h_i \in \mathbb{R}^+ \cup \{0\}$), (iii) the values of these parameters may depend on the order of the model residuals (${}^o h_i \in [0,1]$). In the last two cases, the values of the parameters depend on the obtained residuals. In turn, the residuals depend on $\widetilde{\mathbf{w}}$. Thus, criterion function (6) should only be minimized by iteratively reweighting scenario. Let us denote $\mathbf{w}$, $\mathbf{H}$ and $\mathbf{e}$ in the kth iteration as $\mathbf{w}^{(k)}$, $\mathbf{H}^{(k)}$ and $\mathbf{e}^{(k)}$, respectively. Criterion function (6) for the kth iteration takes the form

$$\begin{aligned} J^{(k)}\left(\widetilde{\mathbf{w}}^{(k)}\right) \triangleq \frac{1}{N}\left(\mathbf{X}\widetilde{\mathbf{w}}^{(k)} - \mathbf{y}\right)^\top \mathbf{H}^{(k)}\left(\mathbf{X}\widetilde{\mathbf{w}}^{(k)} - \mathbf{y}\right) \\ + \tau \left(\widetilde{\mathbf{w}}^{(k)}\right)^\top \widetilde{\mathbf{I}} \widetilde{\mathbf{w}}^{(k)}, \end{aligned} \tag{7}$$

where the elements on the main diagonal of $\mathbf{H}^{(k)} = \operatorname{diag}\left(h_1^{(k)}, h_2^{(k)}, \cdots, h_N^{(k)}\right)$ depend on the residuals from the previous iteration

$$\mathbf{e}^{(k-1)} = \mathbf{X}\widetilde{\mathbf{w}}^{(k-1)} - \mathbf{y}. \tag{8}$$

and take the form

$$h_i^{(k)} = {}^c h_i \cdot {}^l h_i^{(k)} \cdot {}^o h_i^{(k)}. \tag{9}$$

The parameter ${}^{c}h_i$, representing *a priori* confidence to the ith data pair $(\mathbf{x}_i, y_i)$ does not depend on the iteration index k. In contrast, parameter ${}^{l}h_i^{(k)}$ depends on the ith residual from the previous iteration, $(k-1)$th. The following form of ${}^{l}h_i^{(k)}$ is proposed

$${}^{l}h_i^{(k)} = \begin{cases} 0, & e_i^{(k-1)} = 0, \\ \mathcal{L}\left(e_i^{(k-1)}\right) \Big/ \left(e_i^{(k-1)}\right)^2, & e_i^{(k-1)} \neq 0. \end{cases} \tag{10}$$

Indeed, for the quadratic loss function, we obtain ${}^{l}h_i^{(k)} = 1$, for all $i = 1, 2, \cdots, N$; $k = 1, 2, 3, \cdots$. It is well-known from literature [11] that the quadratic (or squared error) loss function does not lead to robustness against noisy data and outliers. A better solution is to use the absolute error function. This loss function is easy to obtain by taking

$${}^{l}h_i^{(k)} = \begin{cases} 0, & e_i^{(k-1)} = 0, \\ 1 \Big/ \left|e_i^{(k-1)}\right|, & e_i^{(k-1)} \neq 0. \end{cases} \tag{11}$$

Many other loss functions may easily be obtained:

- HUBer (HUB) with parameter $\delta > 0$

$${}^{l}h_i^{(k)} = \begin{cases} 1/\delta^2, & \left|e_i^{(k-1)}\right| \leq \delta, \\ 1 \Big/ \left(\delta \left|e_i^{(k-1)}\right|\right), & \left|e_i^{(k-1)}\right| > \delta. \end{cases} \tag{12}$$

- SIGmoidal (SIG) with parameters $\alpha, \beta > 0$

$${}^{l}h_i^{(k)} = \begin{cases} 0, & e_i^{(k-1)} = 0, \\ 1 \Big/ \left(\left(e_i^{(k-1)}\right)^2 \left(1 + \exp\left(-\alpha\left(\left|e_i^{(k-1)}\right| - \beta\right)\right)\right)\right), & e_i^{(k-1)} \neq 0. \end{cases} \tag{13}$$

- SIGmoidal-Linear (SIGL) with parameters $\alpha, \beta > 0$

$${}^{l}h_i^{(k)} = \begin{cases} 0, & e_i^{(k-1)} = 0, \\ 1 \Big/ \left(\left|e_i^{(k-1)}\right| \left(1 + \exp\left(-\alpha\left(\left|e_i^{(k-1)}\right| - \beta\right)\right)\right)\right), & e_i^{(k-1)} \neq 0. \end{cases} \tag{14}$$

- LOGarithmic (LOG)

$${}^{l}h_i^{(k)} = \begin{cases} 0, & e_i^{(k-1)} = 0, \\ \log\left(1 + \left(e_i^{(k-1)}\right)^2\right) \Big/ \left(e_i^{(k-1)}\right)^2, & e_i^{(k-1)} \neq 0. \end{cases} \tag{15}$$

- LOG-Linear (LOGL)

$${}^{l}h_i^{(k)} = \begin{cases} 0, & e_i^{(k-1)} = 0, \\ \log\left(1 + \left(e_i^{(k-1)}\right)^2\right) \Big/ \left|e_i^{(k-1)}\right|, & e_i^{(k-1)} \neq 0. \end{cases} \tag{16}$$

Thus, to minimize the criterion function for the kth iteration the weights are obtained using one of the above equations and the result of optimization of the criterion function from the previous iteration. To start this sequential optimizations, we set the weights in the 0th iteration as ${}^{l}h_i^{(0)} = 1$ for all i. The above minimization problem may be viewed as Iteratively Reweighted Least Square (IRLS) method with the complexity control of the solution.

Let us now explain the meaning of ${}^{o}h_i^{(k)}$ parameters. These parameters depend on the order of the residuals in $(k-1)$th iteration. Let $\pi : \{1, 2, \cdots, N\} \to \{1, 2, \cdots, N\}$ be the permutation function. The rank-ordered residuals satisfy the following conditions:

$$e_{\pi(1)}^{(k-1)} \leq e_{\pi(2)}^{(k-1)} \leq e_{\pi(3)}^{(k-1)} \leq \cdots \leq e_{\pi(N)}^{(k-1)}. \tag{17}$$

For the sake of simplicity, the index of iteration k at the permutation function is temporarily omitted. Now, if ${}^{o}h_i^{(k)}$ parameters fulfill ${}^{o}h_1^{(k)} > {}^{o}h_2^{(k)} > \cdots > {}^{o}h_N^{(k)}$, then it is clear that the impact of outliers is reduced by down-weighting the respective residuals. The disadvantage of this approach is necessity to exchange in each iteration the rows of $\mathbf{X}$ and elements of $\mathbf{y}$ what is a time consuming operation. If we denote the inverse function of $\pi(i)$ as $\pi^{-1}(i)$ then the first term of (7) may be written as

$$\begin{aligned} &\frac{1}{N}\sum_i {}^{c}h_{\pi(i)} \cdot {}^{l}h_{\pi(i)}^{(k)} \cdot {}^{o}h_i^{(k)} \left(e_{\pi(i)}^{(k)}\right)^2 \\ &= \frac{1}{N}\sum_i {}^{c}h_{\pi^{-1}(\pi(i))} \cdot {}^{l}h_{\pi^{-1}(\pi(i))}^{(k)} \cdot {}^{o}h_{\pi^{-1}(i)}^{(k)} \left(e_{\pi^{-1}(\pi(i))}^{(k)}\right)^2. \end{aligned} \tag{18}$$

Using the identity $\pi^{-1}(\pi(i)) = i$ the above sum equals

$$\begin{aligned} &\frac{1}{N}\sum_i {}^{c}h_i \cdot {}^{l}h_i^{(k)} \cdot {}^{o}h_{\pi^{-1}(i)}^{(k)} \left(e_i^{(k)}\right)^2 \\ &= \frac{1}{N}\sum_i {}^{c}h_i \cdot {}^{l}h_i^{(k)} \cdot {}^{o}\breve{h}_i^{(k)} \left(e_i^{(k)}\right)^2 \end{aligned} \tag{19}$$

where ${}^{o}\breve{h}_i^{(k)} = {}^{o}h_{\pi^{-1}(i)}^{(k)}$. In sequel, we obtain

$${}^{o}\breve{h}_{\pi(i)}^{(k)} = {}^{o}h_{\pi(\pi^{-1}(i))}^{(k)} = {}^{o}h_i^{(k)}. \tag{20}$$

Finally, if we denote the permutation function for kth iteration as $\pi^{(k)}(i)$, then according to the above result, (9) should be replaced by

$$h_i^{(k)} = {}^{c}h_i \cdot {}^{l}h_i^{(k)} \cdot {}^{o}\breve{h}_{\pi^{(k-1)}(i)}^{(k)}. \tag{21}$$

The form of parameters ${}^{o}h_i^{(k)}$ is proposed to be piecewise-linear

$${}^{o}h_i^{(k)} = \{[(c-i)/(2\xi) + 1/2] \wedge 1\} \vee 0 \tag{22}$$

or sigmoidal

$${}^{o}h_i^{(k)} = 1/(1 + \exp(a(i-c))), \tag{23}$$

where $\wedge$ and $\vee$ denotes *min* and *max* operations, respectively. Both functions, which may be called the weighting functions, are nonincreasing with respect to argument $i \in \{1, 2, \cdots, N\}$. For $i = c$ these functions are equal to 0.5. Parameters $\xi > 0$ and $a > 0$ influence a slope. In the case of piecewise-linear function, for $i \in [c-\xi, c+\xi]$ its value linearly decreases from 1 to 0. For sigmoidal function, for $i \in [c-2.944/a, c+2.944/a]$ its value decreases from 0.95 to 0.05. In the rest of the work, the functions defined by (22) and (23) are called

Sigmoidally-weighted OWA (SOWA) and Piecewise-Linearly-weighted OWA (PLOWA), respectively. If ordering of residuals is not used, which is equivalent to using uniformly weighting function for OWA (for all i,k: ${}^{o}h_i^{(k)}=1$), then we call this case regression without ordering (or with none weighting function).

3. ϵ-insensitive loss functions

It is well known in machine learning that too precise learning on a training set can lead to the so-called overfitting, and in consequence to poor generalization ability on data point previously unseen [7]. Tolerating a small errors in fitting on a given dataset, can improve correctness on the test dataset [13]. Motivated by the results of statistical learning theory, Vapnik introduced the ϵ-insensitive loss function. This function disregards errors below some $\epsilon>0$, chosen a priori:

$$\mathcal{L}(\zeta)=\begin{cases}0, & |\zeta|\leq\epsilon,\\ |\zeta|-\epsilon, & |\zeta|>\epsilon.\end{cases} \tag{24}$$

Various ϵ-insensitive loss functions may be considered, including ϵ-insensitive quadratic, ϵ-insensitive Huber, and so on. Let us start our consideration from the ϵ-insensitive quadratic loss

$$\mathcal{L}(\zeta)=\begin{cases}0, & |\zeta|-\epsilon\leq 0,\\ (\epsilon-\zeta)^2, & \epsilon-\zeta<0,\\ (\epsilon+\zeta)^2, & \epsilon+\zeta<0.\end{cases} \tag{25}$$

Taking into account the above equation, the first term of (5), assuming $h_i=1$ for all $i=1,2,\cdots,N$, may be written as

$$\begin{aligned}\sum_{i=1}^{N}\mathcal{L}\left(\widetilde{\mathbf{w}}^{\top}\widetilde{\mathbf{x}}_i-y_i\right)&=\sum_{i=1}^{N}h_i^{+}\left(-\widetilde{\mathbf{w}}^{\top}\widetilde{\mathbf{x}}_i+y_i+\epsilon\right)^2\\ &+\sum_{i=1}^{N}h_i^{-}\left(\widetilde{\mathbf{w}}^{\top}\widetilde{\mathbf{x}}_i-y_i+\epsilon\right)^2,\end{aligned} \tag{26}$$

where h_i^{+} (h_i^{-}) are equal to zero for $-\widetilde{\mathbf{w}}^{\top}\widetilde{\mathbf{x}}_i+y_i+\epsilon\geq 0$ ($\widetilde{\mathbf{w}}^{\top}\widetilde{\mathbf{x}}_i-y_i+\epsilon\geq 0$) and 1 otherwise. Thus, the ϵ-insensitive quadratic loss function may be decomposed into two asymmetric quadratic loss functions. Let $\mathbf{X}_e$ be the $2N\times(t+1)$ matrix

$$\mathbf{X}_e^{\top}\triangleq\left[\mathbf{X}^{\top},-\mathbf{X}^{\top}\right] \tag{27}$$

and $\mathbf{y}_e$ be the $2N$-dimensional vector $\mathbf{y}_e^{\top}=\left[\mathbf{y}^{\top}-\epsilon\mathbf{1}^{\top},-\mathbf{y}^{\top}-\epsilon\mathbf{1}^{\top}\right]$. Vector $\mathbf{1}$ denotes the vector with all entries equal to 1. Using the above mentioned notation, criterion function (7) for kth iteration takes the form

$$\begin{aligned}J^{(k)}\left(\widetilde{\mathbf{w}}^{(k)}\right)\triangleq\frac{1}{N}&\left(\mathbf{X}_e\widetilde{\mathbf{w}}^{(k)}-\mathbf{y}_e\right)^{\top}\mathbf{H}^{(k)}\left(\mathbf{X}_e\widetilde{\mathbf{w}}^{(k)}-\mathbf{y}_e\right)\\ &+\tau\left(\widetilde{\mathbf{w}}^{(k)}\right)^{\top}\widetilde{\mathbf{I}}\widetilde{\mathbf{w}}^{(k)},\end{aligned} \tag{28}$$

where the elements on the main diagonal of $\mathbf{H}^{(k)}$ (now, $(2N)\times(2N)$ matrix) depend on residuals from the previous iteration

$$\mathbf{e}^{(k-1)}=\mathbf{X}_e\widetilde{\mathbf{w}}^{(k-1)}-\mathbf{y}_e. \tag{29}$$

The fitting of the ith data pair is represented by the ith and the $(i+N)$th element of $\mathbf{e}$. If both $e_i^{(k)}$ and $e_{i+N}^{(k)}$ are greater than or equal to zero, then the ith datum falls in the kth iteration into the insensibility zone. If $e_i^{(k)}$ ($e_{i+N}^{(k)}$) is less than zero, then the ith datum is below (above) the insensibility zone in the kth iteration and should be penalized. For the ϵ-insensitive quadratic loss we have

$${}^{l}h_i^{(k)}=\begin{cases}0, & e_i^{(k-1)}\geq 0,\\ 1, & e_i^{(k-1)}<0.\end{cases} \tag{30}$$

Many other ϵ-insensitive loss functions may easily be obtained:

- VAPnik (VAP)

$${}^{l}h_i^{(k)}=\begin{cases}0, & e_i^{(k-1)}\geq 0,\\ -1\Big/e_i^{(k-1)}, & e_i^{(k-1)}<0.\end{cases} \tag{31}$$

- HUBer (ϵHUB) with parameter $\delta>0$

$${}^{l}h_i^{(k)}=\begin{cases}0, & e_i^{(k-1)}\geq 0,\\ 1/\delta^2, & 0>e_i^{(k-1)}\geq-\delta,\\ -1\Big/\left(\delta\left|e_i^{(k-1)}\right|\right), & e_i^{(k-1)}<-\delta.\end{cases} \tag{32}$$

- SIGmoidal (ϵSIG) with parameters $\alpha,\beta>0$

$${}^{l}h_i^{(k)}=\begin{cases}0, & e_i^{(k-1)}\geq 0,\\ 1\Big/\left(\left(e_i^{(k-1)}\right)^2\left(1+\exp\left(\alpha\left(e_i^{(k-1)}+\beta\right)\right)\right)\right), & e_i^{(k-1)}<0.\end{cases} \tag{33}$$

- SIGmoidal-Linear (ϵSIGL) with parameters $\alpha,\beta>0$

$${}^{l}h_i^{(k)}=\begin{cases}0, & e_i^{(k-1)}\geq 0,\\ -1\Big/\left(e_i^{(k-1)}\left(1+\exp\left(\alpha\left(e_i^{(k-1)}+\beta\right)\right)\right)\right), & e_i^{(k-1)}<0.\end{cases} \tag{34}$$

- LOGarithmic (ϵLOG)

$${}^{l}h_i^{(k)}=\begin{cases}0, & e_i^{(k-1)}\geq 0,\\ \log\left(1+\left(e_i^{(k-1)}\right)^2\right)\Big/\left(e_i^{(k-1)}\right)^2, & e_i^{(k-1)}<0.\end{cases} \tag{35}$$

- LOG-Linear (ϵLOGL)

$${}^{l}h_i^{(k)}=\begin{cases}0, & e_i^{(k-1)}\geq 0,\\ -\log\left(1+\left(e_i^{(k-1)}\right)^2\right)\Big/e_i^{(k-1)}, & e_i^{(k-1)}<0.\end{cases} \tag{36}$$

Our *a priori* confidence to the ith datum (${}^{c}h_i\in[0,1]$) should be 'doubled', i.e., ${}^{c}h_{i+N}={}^{c}h_i$, for $i=1,2,\cdots,N$, because every datum in criterion function (28) is also doubled. Moreover, ${}^{o}h_i$ parameters should be obtained in a different way. Let a distance from the insensibility zone be

$s_i^{(k-1)} = -(e_i^{(k-1)} \wedge e_{i+N}^{(k-1)} \wedge 0)$ for $i = 1, 2, \cdots, N$ and the permutation function for the rank-ordered s_is be $\pi^{(k-1)}$ for $(k-1)$th iteration. Let $N_\gamma^{(k-1)}$ denotes the number of s_is equal to zero, i.e., $s_{\pi^{(k-1)}(i)} = 0$ for $i = 1, 2, \cdots, N_\gamma^{(k-1)}$. Finally, for the kth iteration ${}^o\breve{h}^{(k)}_{\pi^{(k-1)}(i)} = 1$ for $i = 1, 2, \cdots, N_\gamma^{(k-1)}$ and takes a form of piecewise-linear (22) or sigmoidal (23) function for $i = N_\gamma^{(k-1)} + 1, N_\gamma^{(k-1)} + 2, \cdots, N$.

4. A method of solution

In the previous two sections it has been shown that linear regression problems with various loss functions (including the ϵ-insensitive ones), regularization, and ordered weighted averaging of the residuals may be formulated as an iteratively reweighted least square scenario. For the kth iteration, we need to minimize criterion function (7) or in the case of an ϵ-insensitive loss function (28). In this section, we focus on minimization of (7). Criterion (28) has the same form after replacing $\mathbf{X}$ by $\mathbf{X}_e$ and $\mathbf{y}$ by $\mathbf{y}_e$.

The optimality condition for the kth iteration is obtained by differentiating (7) with respect to $\widetilde{\mathbf{w}}$ and setting the result equals to zero

$$\widetilde{\mathbf{w}}^{(k)} = \left(\mathbf{X}^\top \mathbf{H}^{(k)} \mathbf{X} + \tau N \widetilde{\mathbf{I}}\right)^{-1} \mathbf{X}^\top \mathbf{H}^{(k)} \mathbf{y}. \tag{37}$$

The procedure of iteratively reweighted least square error minimization for linear regression can be summarized in the following steps:

1. Fix $\tau > 0$ and $\mathbf{H}^{(0)} = \mathbf{I}$. Set the iteration index $k = 0$.
2. $\widetilde{\mathbf{w}}^{(k)} = \left(\mathbf{X}^\top \mathbf{H}^{(k)} \mathbf{X} + \tau N \widetilde{\mathbf{I}}\right)^{-1} \mathbf{X}^\top \mathbf{H}^{(k)} \mathbf{y}$.
3. $\mathbf{e}^{(k)} = \mathbf{X}\widetilde{\mathbf{w}}^{(k)} - \mathbf{y}$.
4. $\mathbf{H}^{(k+1)} = \text{diag}\left(h_1^{(k+1)}, h_2^{(k+1)}, \cdots, h_N^{(k+1)}\right)$, where each $h_i^{(k+1)}$, for $i = 1, 2, \cdots, N$ is obtained by (21) and depends on the selected loss function and the type of weighting function: (22) or (23).
5. if $k > 1$ and $\left\|\widetilde{\mathbf{w}}^{(k)} - \widetilde{\mathbf{w}}^{(k-1)}\right\|_2 < \xi$, then stop else $k \leftarrow k + 1$, go to (2).

Remarks. The iterations are stopped as soon as the Euclidean norm in a successive pair of $\widetilde{\mathbf{w}}$ vectors is less than ξ. The quantity ξ is a pre-set small positive value. In all experiments $\xi = 10^{-3}$ is used. The above algorithm requires the inversion of an $(t+1) \times (t+1)$ matrix that leads to a running time of $\mathcal{O}\left((t+1)^3\right)$ where t stands for the dimensionality of input data. Thus, this algorithm is computationally infeasible for large dimensionality of the data.

In the above algorithm, to solve unconstrained quadratic optimization problem (7) the well-known conjugate gradient approach can be used [20]. In contrast to solution (37), this algorithm produces a minimizing sequence $\widetilde{\mathbf{w}}^{(k),[n]}$, where $n = 0, 1, \cdots$. For the sake of simplicity superscript (k) is omitted

$$\widetilde{\mathbf{w}}^{[n+1]} = \widetilde{\mathbf{w}}^{[n]} + \nu^{[n]} \mathbf{d}^{[n]}, \tag{38}$$

where $\nu^{[n]}$, $\mathbf{d}^{[n]}$ denote the step size and the search direction for the nth iteration of the conjugate gradient, respectively. After some simple algebra, the criterion function (7) may be expressed as

$$J\left(\widetilde{\mathbf{w}}\right) = \frac{1}{2} \widetilde{\mathbf{w}}^\top \mathbf{G} \mathbf{w} - \mathbf{b}^\top \widetilde{\mathbf{w}} + c, \tag{39}$$

where $\mathbf{G} = \frac{2}{N} \mathbf{X}^\top \mathbf{H}\, \mathbf{X} + 2\tau \widetilde{\mathbf{I}}$, $\mathbf{b} = \frac{2}{N} \mathbf{X}^\top \mathbf{H} \mathbf{y}$ and $c = \frac{1}{N} \mathbf{y}^\top \mathbf{H} \mathbf{y}$. Let us assume that the search direction is known, then the step size is chosen to minimize $J\left(\widetilde{\mathbf{w}}^{[n+1]}\right) = J\left(\widetilde{\mathbf{w}}^{[n]} + \nu^{[n]} \mathbf{d}^{[n]}\right)$. Differentiating above with respect to $\nu^{[n]}$ and setting the result equals to zero, we have

$$\nu^{[n]} = -\frac{2\tau \left(\mathbf{d}^{[n]}\right)^\top \widetilde{\mathbf{I}} \widetilde{\mathbf{w}}^{[n]} + \frac{2}{N} \left(\mathbf{d}^{[n]}\right)^\top \mathbf{X}^\top \mathbf{H}\, \mathbf{e}^{[n]}}{\left(\mathbf{d}^{[n]}\right)^\top \mathbf{G} \mathbf{d}^{[n]}}, \tag{40}$$

where $\mathbf{e}^{[n]} = \mathbf{X}\widetilde{\mathbf{w}}^{[n]} - \mathbf{y}$.

In the conjugate gradient method, the current search direction should be $\mathbf{G}$-conjugate to the previously chosen directions, i.e. $\left(\mathbf{d}^{[n_1]}\right)^\top \mathbf{G} \mathbf{d}^{[n_2]} = 0$ for all $n_1 \neq n_2$. A new search direction is obtained as a combination of the previous one and the current gradient vector $\mathbf{g}^{[n]}$

$$\mathbf{d}^{[n]} = \mathbf{g}^{[n]} + \beta^{[n]} \mathbf{d}^{[n-1]}, \tag{41}$$

where $\beta^{[n]}$ is chosen to obtain $\mathbf{G}$-conjugacy with the previous direction.
Thus, $\left(\mathbf{d}^{[n-1]}\right)^\top \mathbf{G} \left(\mathbf{g}^{[n]} + \beta^{[n]} \mathbf{d}^{[n-1]}\right) = 0$. After some simple algebra, we have

$$\beta^{[n]} = -\frac{\left(\mathbf{d}^{[n-1]}\right)^\top \mathbf{G} \mathbf{g}^{[n]}}{\left(\mathbf{d}^{[n-1]}\right)^\top \mathbf{G} \mathbf{d}^{[n-1]}}. \tag{42}$$

The gradient vector is obtained using (7)

$$\mathbf{g}^{[n]} = \left.\frac{\partial J\left(\widetilde{\mathbf{w}}\right)}{\partial \mathbf{w}}\right|_{\widetilde{\mathbf{w}} = \widetilde{\mathbf{w}}^{[n]}} = 2\tau \widetilde{\mathbf{I}} \widetilde{\mathbf{w}}^{[n]} + \frac{2}{N} \mathbf{X}^\top \mathbf{H} \left(\mathbf{X}\widetilde{\mathbf{w}}^{[n]} - \mathbf{y}\right). \tag{43}$$

Comparing (40) and (43), a simpler form of the step size is obtained

$$\nu^{[n]} = -\frac{\left(\mathbf{d}^{[n]}\right)^\top \mathbf{g}^{[n]}}{\left(\mathbf{d}^{[n]}\right)^\top \mathbf{G} \mathbf{d}^{[n]}}. \tag{44}$$

The minimization of (7) using the conjugate gradient method may be summarized in the following steps

1. Set the iteration index $n = 0$ and $\widetilde{\mathbf{w}}^{[0]} = \mathbf{0}$.
2. Calculate gradient vector $\mathbf{g}^{[n]}$ using (43).
3. if $n = 0$, then $\beta^{[0]} = 0$, else calculate $\beta^{[n]}$ using (42).
4. Calculate search direction $\mathbf{d}^{[n]}$ using (41).
5. Calculate step size $\nu^{[n]}$ using (44).
6. Update $\widetilde{\mathbf{w}}$ using (38).
7. if $\left\|\widetilde{\mathbf{w}}^{[n+1]} - \widetilde{\mathbf{w}}^{[n]}\right\|_2 < \zeta$, then stop, else $n \leftarrow n + 1$, go to (2).

Remarks. The iterations are stopped as soon as the Euclidean norm in a successive pair of $\widetilde{\mathbf{w}}$ vectors is less than ζ where ζ is a pre-set small positive value. In all experiments $\zeta = 10^{-5}$ is used. The conjugate gradient algorithm converges theoretically in $p+1$ steps where p is rank of matrix $\mathbf{G}$. If $\mathbf{G}$ is full rank then the algorithm converges in $t+1$ steps where t denotes the dimensionality of input data. This algorithm replaces step (2) in the previous algorithm. The quantity $\left(\mathbf{d}^{[n]}\right)^{\top} \mathbf{G}\mathbf{d}^{[n]}$ calculated in step (5) may be saved and used in step (3) of the next iteration. If the above algorithm replaces step (2) from the algorithm presented at the beginning of this section, then the Generalized Ordered Linear Regression with Regularization (GOLRR) method is obtained.

5. Numerical experiments and discussion

All experiments were done on Hewlett-Packard HP Compaq dx7300 Intel Core 2 CPU 6300 @ 1.86 GHz with 1GB RAM, running Windows XP (Service Pack 2) and MATLAB 6.5 environment. The following values of coefficients were used for loss functions: $\alpha = 8$, $\beta = 1$, $\delta = 0.5$, and for weighting functions (22), (23): $c = 0.6$, $\xi = 0.2$, $a = 0.2$. In all experiments we assume no a priori knowledge about data pairs, i.e. ${}^{c}h_i = 1$, $i = 1, 2, \cdots, N$.

5.1. Simple linear regression for data with outliers. The purpose of experiments in this subsection was to evaluate the performance of the proposed method when applied with various loss functions and types of weighting function to simple linear data with noise and outliers. The dataset used in those experiments was generated in the following way. We simulated 100 datasets, each having 100 observations. First, for each dataset, 100 random uniformly distributed input data (in the range from 0.0 to 50.0) were generated, x_i; $i = 1, 2, \cdots, 100$. After sorting these observations, the output data were generated as $y_i = 1.5x_i + \varsigma_i$ for $i = 1, 2, \cdots, 49, 81, \cdots, 100$ (69 true observations) and $y_i = 1.0x_i + \varsigma_i$ for $i = 50, \cdots, 80$ (31 outliers). Each ς_i (noise) was generated as a sum of 4 independent random uniformly distributed variables in $[-1, 1]$. Thus, the true vector of the model coefficients was $\mathbf{w}^{\top} = [w_1, w_0] = [1.5, 0]$.

Table 1 shows the results obtained for each combination of the loss function and the weighting function: the mean value $\pm$ standard deviation of the coefficients and the computation time in seconds.

The obtained results show that for all loss function used, the mean values of parameter w_1 are closer to the real value if a weighting function is used. In four cases (SQR, HUB, LOG, LOGL), the PLOWA performs better than the SOWA. In two cases (LIN, SIGL), the same mean values of w_1 were obtained for the SOWA and the PLOWA, and in the case of SIG loss function the results were better for the SOWA. The values of the standard deviation for parameter w_1 were very similar for almost all loss functions and all weighting functions (except SQR and LOGL with no weighting function used). In these cases the value of the standard deviation for parameter w_1 was twice as big as for all other combinations of loss and weighting functions. The best results, with respect to parameter w_1, were obtained for LOG loss function with the PLOWA and the worst ones for SQR loss function without weighting function (the traditional OLS).

Table 1
The influence of the loss and the weighting functions on the proposed method performance on the first dataset. The true values of the coefficients are $w_1 = 1.5$ and $w_0 = 0$

Loss function		Weighting function		
		None	SOWA	PLOWA
SQR	w_1	1.305 ± 0.025	1.482 ± 0.011	1.488 ± 0.010
	w_0	-0.085 ± 0.610	-0.116 ± 0.301	-0.164 ± 0.294
	Time	0.125	0.219	0.188
LIN	w_1	1.452 ± 0.017	1.496 ± 0.014	1.496 ± 0.013
	w_0	0.093 ± 0.378	-0.150 ± 0.389	-0.136 ± 0.385
	Time	0.359	0.375	0.375
HUB	w_1	1.456 ± 0.013	1.496 ± 0.013	1.497 ± 0.013
	w_0	0.110 ± 0.341	0.009 ± 0.356	0.008 ± 0.354
	Time	0.359	0.328	0.313
SIG	w_1	1.495 ± 0.012	1.498 ± 0.013	1.497 ± 0.012
	w_0	0.025 ± 0.288	0.008 ± 0.352	0.013 ± 0.344
	Time	0.219	1.562	0.234
SIGL	w_1	1.444 ± 0.013	1.497 ± 0.012	1.497 ± 0.012
	w_0	0.111 ± 0.331	0.011 ± 0.348	0.003 ± 0.334
	Time	0.656	1.562	1.281
LOG	w_1	1.483 ± 0.010	1.497 ± 0.011	1.498 ± 0.011
	w_0	0.026 ± 0.287	0.011 ± 0.320	0.007 ± 0.317
	Time	0.250	0.256	0.266
LOGL	w_1	1.316 ± 0.026	1.490 ± 0.010	1.493 ± 0.009
	w_0	0.516 ± 0.591	-0.062 ± 0.287	-0.084 ± 0.275
	Time	0.219	0.187	0.203

As far as the mean values of parameter w_0 are concerned, the use of SQR and LIN loss functions lead to better results if no weighting function is applied. For SIG and LOGL loss functions, the SOWA gives better results, and for the HUB, SIGL and LOG loss functions, the PLOWA performs better. It should be noted that the best mean values of w_0 obtained for SQR, LIN and LOGL loss functions are more than eight-fold bigger than the worst case among the best mean values for the other four loss functions (HUB, SIG, SIGL, LOG). The values of the standard deviation for parameter w_0 were very similar for all loss functions and all weighting functions except SQR and LOGL loss functions with no weighting function used. In these cases, the value of the standard deviation for parameter w_0 was once again twice as big as for all other combinations of loss and weighting functions. The best results, with respect to parameter w_0, were obtained for the SIGL loss function with the PLOWA and the worst results were obtained for the LOGL loss function without weighting function.

The computing time for all combinations of loss and weighting functions is very small; usually 3-6 times greater with respect to the OLS. Only in three cases (SIGL/SOWA, SIGL/PLOWA and HUB/SOWA) computing time exceeds one second (about 10 times greater with respect to the OLS). An example of the results obtained in this set of experiments is presented in Fig. 1.

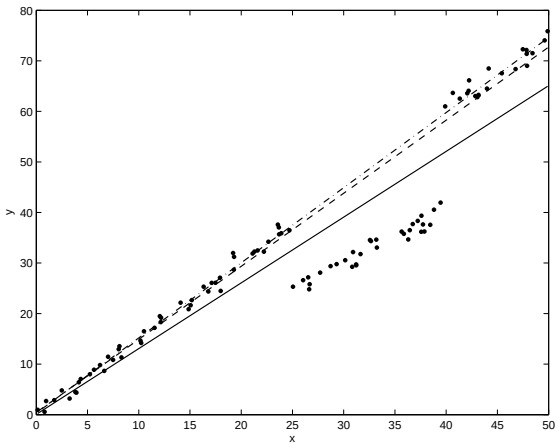

Fig. 1. An example of data from the first dataset, and the regression lines obtained by: squared loss without ordering (solid line), Huber loss without ordering (dashed line), logarithmic loss and piecewise-linear weighting function (dash-dot line)

5.2. Simple linear regression for data with background noise The purpose of experiments in this subsection was to evaluate the performance of the proposed method when applied with various loss and ordering functions to simple linear data with background noise. The dataset used in these experiments consists of 100 simulated, smaller datasets, each having 1000 observations, generated in the following way. First, for each dataset, 1000 random uniformly distributed input data (in the range from 0.0 to 50.0) were generated, x_i; $i = 1, 2, \cdots, 1000$. The output data were generated as $y_i = 0.5x_i + 7 + \varsigma_i$. The ς_i (noise) was standard normal (zero mean and variance equal to one). Thus, the true vector of the model coefficients was $\mathbf{w}^\top = [w_1, w_0] = [0.5, 7]$. To these data, 2000 points (x_i, y_i) $i = 1001, \cdots, 3000$ were added. Coordinates of these data points were random, uniformly distributed in the ranges $[0, 50]$ and $[0, 35]$, respectively.

Table 2 shows the results obtained for each combination of the loss function and the weighting function: the mean values $\pm$ standard deviation of the coefficients, and computation time in seconds. In these experiments, the data points (x_i, y_i); $i = 1, 2, \cdots, 2000$ were used. Thus, we have 1000 observations taken from the known linear model and 1000 points of background noise.

In this second set of experiments, the obtained results show that for all loss functions used, the mean values of parameter w_1 are closer to the real value if a weighting function is used. In three cases of loss functions (SQR, SIGL, LOGL), the PLOWA performs better than the SOWA. In three cases (HUB, SIG, LOG) the same mean values of w_1 were obtained for the SOWA and the PLOWA, and in the case of LIN loss function the results were better for the SOWA. The values of the standard deviation for parameter w_1 were the highest for the SIG loss function. In the cases of LIN/PLOWA, SQR/Non and LOGL/Non combinations, the mean value for parameter w_1 is greater than for all others cases. The best results, with respect to parameter w_1, were obtained for HUB loss function with the SOWA and the worst results were obtained for SQR loss function without weighting function.

Table 2
The influence of the loss and the weighting functions on the proposed method performance on the second dataset. The true values of the coefficients are $w_1 = 0.5$ and $w_0 = 7.0$

Loss function		Weighting function		
		None	SOWA	PLOWA
SQR	w_1	0.251 ± 0.012	0.488 ± 0.003	0.492 ± 0.004
	w_0	12.235 ± 0.342	7.259 ± 0.100	7.165 ± 0.108
	Time	0.188	0.625	0.594
LIN	w_1	0.456 ± 0.011	0.490 ± 0.011	0.487 ± 0.021
	w_0	7.928 ± 0.238	7.217 ± 0.255	7.270 ± 0.448
	Time	1.172	2.375	2.172
HUB	w_1	0.461 ± 0.004	0.498 ± 0.003	0.498 ± 0.003
	w_0	7.813 ± 0.120	7.052 ± 0.091	7.045 ± 0.091
	Time	0.765	1.187	1.187
SIG	w_1	0.481 ± 0.025	0.484 ± 0.029	0.484 ± 0.023
	w_0	7.428 ± 0.552	7.381 ± 0.618	7.369 ± 0.505
	Time	0.704	1.015	0.984
SIGL	w_1	0.443 ± 0.005	0.497 ± 0.006	0.498 ± 0.005
	w_0	8.235 ± 0.135	7.063 ± 0.176	7.054 ± 0.170
	Time	0.641	0.968	0.922
LOG	w_1	0.478 ± 0.003	0.495 ± 0.003	0.495 ± 0.003
	w_0	7.480 ± 0.104	7.132 ± 0.086	7.136 ± 0.087
	Time	0.656	1.032	1.000
LOGL	w_1	0.345 ± 0.010	0.493 ± 0.003	0.495 ± 0.003
	w_0	10.246 ± 0.279	7.144 ± 0.093	7.108 ± 0.101
	Time	0.578	0.735	0.719

In the case of the mean values of parameter w_0, as well as in the case of w_1, application of a weighting function leads to better results. For LIN and LOG loss functions, the SOWA gives better results, and for the other loss functions the PLOWA performs better. It should be noted that the best mean values of parameter w_0 are obtained for exactly the same loss/weighting functions combinations, which assured the best mean values of parameter w_1. The best mean value of w_0 is obtained for the HUB loss function and the PLOWA. The worst mean value was obtained for the SQR/None combination; the second worst results were obtained for LOGL loss function without weighting function. The values of the standard deviation for parameter w_0 were very similar for almost all loss functions and all weighting functions (except SIG loss function, and SQR, LOGL loss functions with no weighting function used). The best results, with respect to parameter w_0, were obtained for the HUB loss function with the PLOWA, and the worst results were obtained for the SQR loss function without weighting function. The computing time for all combinations of loss and weighting functions is very small, hence, in most cases, more important than in the first set of experiments. In six cases the computing time exceeds one second and in one case (LIN/SOWA) it is greater than two seconds. An example of the results obtained in this set of experiments is presented in Fig. 2.

The next experiment was dedicated to evaluate the influence of the growing number of outliers on the proposed method. As in the above experiments, data points (x_i, y_i) for $i = 1, 2, \cdots, 1000 + 100 \cdot \rho$ were used. Thus, we have 1000 observations taken from the known linear model and $100 \cdot \rho$

points of background noise, where $\rho = 1, 2, \cdots, 20$. thus, the number of background noise points was varied from 100 to 2000 (with the step 100). The performance of the method was evaluated by means of the mean squared error (of the difference between the true and the obtained values of the model parameters). The results obtained are presented in Fig. 3 for the following cases: the squared loss without any weighting function (the line with the triangle markers), the Huber loss without any weighting function (the line with the cross markers), the logarithmic loss and the PLOWA (the line with the square markers). Of course, for the squared loss function, the growing number of outliers has the greatest impact on the model quality. For the Huber loss function the impact is similar although not so great. Applying the logarithmic loss function and the PLOWA, we do not observe a strong deterioration of the model quality, even if the number of outliers is twice as large as the number of observations!

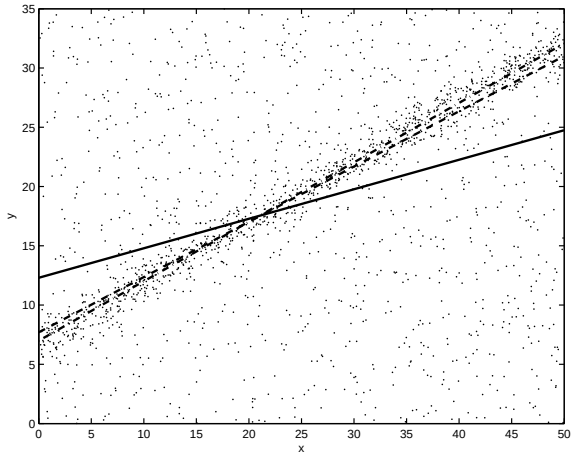

Fig. 2. An example of data from the second dataset, and the regression lines obtained by: squared loss without ordering (solid line), Huber loss without ordering (dashed line), logarithmic loss and piecewise-linear weighting function (dash-dot line)

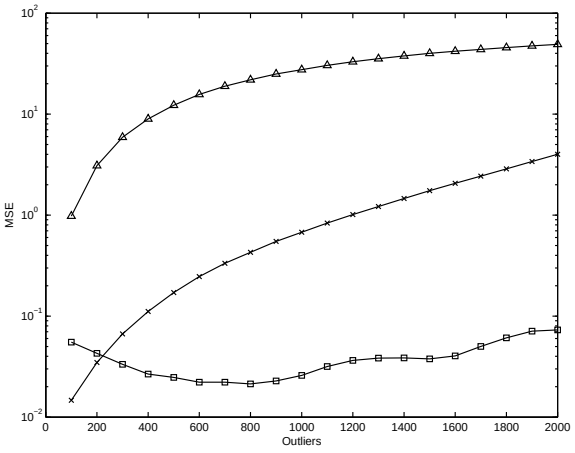

Fig. 3. Comparison of the quality of regression lines for a growing number of background noise (outliers) obtained by: squared loss without ordering (line with triangle markers), Huber loss without ordering (line with cross markers), logarithmic loss and piecewise-linear weighting function (line with square markers). Vertical axis is in logarithmic scale

5.3. Multiple linear regression with ϵ-insensitive loss function and ℓ_2 regularization. The purpose of experiments in this subsection was to investigate the generalization ability of a model obtained by the proposed method with the ϵ-insensitive loss function and regularization when applied to an ill-posed regression problem. The dataset used in these experiments consist of 100 simulated, smaller datasets, each having 50 observations generated in the following way. First, for each dataset, $11 \cdot 50$ random, uniformly distributed input data in the range from -10.0 to 10.0 were generated, $x_{j,i}$; $j = 1, 2, \cdots, 11$; $i = 1, 2, \cdots, 50$. The output data were generated as $y_i = -2.5x_{1,i} - 2.0x_{2,i} - 1.5x_{3,i} - 1.0x_{4,i} - 0.5x_{5,i} - 0.0x_{6,i} + 0.5x_{7,i} + 1.0x_{8,i} + 1.5x_{9,i} + 2.0x_{10,i} + 2.5x_{11,i} + 10 + \varsigma_i$. Each ς_i (noise) was generated as a sum of 2 independent random uniformly distributed variables in $[-1, 1]$ and $[-1.5, 1.5]$. Thus, the true vector of the model coefficients was $\mathbf{w}_t^\top = [-2.5, -2.0, -1.5, -1.0, -0.5, 0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 10.0]$. Each dataset was divided into the learning ($i = 1, 2, \cdots, 7$) and the testing part ($i = 8, 9, \cdots, 50$).

In the experiments, the squared ϵ-insensitive loss function was used. The insensitivity parameter ϵ was varied from 0.2 to 4.0 (with the step 0.2). The regularization parameter τ was changed from 0.00025 to 0.0125 (with the step 0.00025). The models were created using the training parts of the datasets. The generalization ability was evaluated using the testing parts. Figure 4 presents the contour plot of the squared sum of errors on the testing parts vs. ϵ and τ. The darker the area the greater the generalization ability of the model. The main conclusion from these experiments is as follows: for values $\epsilon = 2.9$ and $\tau = 0.0110$, so different from zero, the model has the greatest generalization ability. Similar results were obtained for the other loss functions, but due to the volume of the work, they were not presented here.

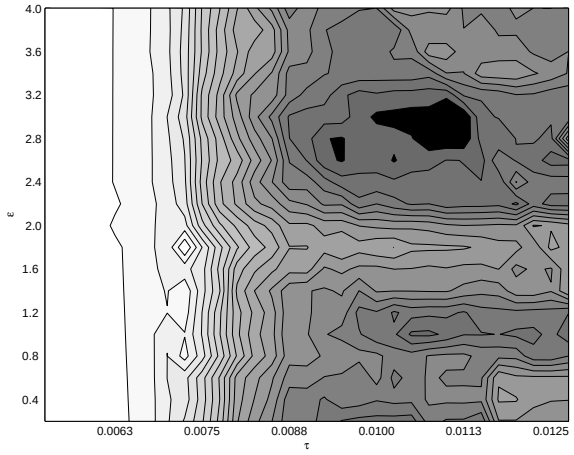

Fig. 4. The generalization ability of the linear model as the function of insensitivity parameter ϵ and regularization parameter τ. The darker the area the greater the generalization ablity of the model

6. Conclusions

In this paper it has been shown that linear regression for various loss functions, ordered weighted averaging of residuals,

and regularization, can be formulated as minimization of the iteratively reweighted least squares (IRLS) criterion function. The proposed criterion function may easily be extended for the ϵ-insensitive loss functions. The conjugate gradient algorithm is a computationally effective method for minimization of the proposed criterion function. An experimental analysis on synthetic datasets shows that the proposed method is high-breakdown robust to outliers. Depending on the selected loss functions, weighting vector for ordered residuals, the well known classical regression method can be obtained, as well as many new methods, including for example: the least median with the Huber loss, the trimmed ϵ-insensitive absolute deviation with regularization. Regardless of a loss function and a weighting vector used, the numerical solution to the regression problem can be obtained using the same algorithm based on the iteratively reweighted least squares scenario with conjugate gradient optimization.

REFERENCES

[1] A.M. Legendre, *Nouvelles Methodes Pour la Determination des Orbites des Cometes*, Didot, Paris, 1805.

[2] K.F. Gauss, *Theory of the Motion of the Heavenly Bodies Moving About the Sun in Conic Sections: a translation of Gauss's Theoria Motus*, Little, Brown and Company, Boston, 1857.

[3] R. Deutsch, *Estimation Theory*, Prentice-Hall, Englewood Cliffs, 1965.

[4] J.M. Łęski, "ϵ-insensitive fuzzy c-regression models: Introduction to ϵ-insensitive fuzzy modeling", *IEEE Trans. Syst., Man and Cybern. – Part B: Cybern.* 34 (1), 4–15 (2004).

[5] J.M. Łęski, "TSK-fuzzy modeling based on ϵ-insensitive learning", *IEEE Trans. Fuzzy Syst.* 13 (2), 181–193 (2005).

[6] J.M. Łęski, "Iteratively reweighted least squares classifier and its ℓ_2- and ℓ_1-regularized kernel versions", *Bull. Pol. Ac.: Tech.* 58 (1), 171–182 (2010).

[7] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1995.

[8] J. Dziekan, W. Matczak, and J. Korbicz, "Active fault-tolerant control design for Takagi-Sugeno fuzzy systems", *Bull. Pol. Ac.: Tech.* 59 (1), 93–102 (2011).

[9] M. Kaminski and T. Orlowska-Kowalska, "Optimisation of neural state variables estimators of two-mass drive system using the Bayesian regularization method", *Bull. Pol. Ac.: Tech.* 59 (1), 33–38 (2011).

[10] P.J. Huber, "Robust estimation of location parameter", *Ann. Math. Stat.* 35 (1), 73–101 (1964).

[11] P.J. Huber, *Robust Statistics*, Wiley, New York, 1981.

[12] P.J. Rousseeuw and A.M. Leroy, *Robust Regression and Outliers Detection*, John Wiley, New York, 1987.

[13] V. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.

[14] R.R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decision making", *IEEE Trans. Syst., Man Cybern.* 18(1), 183-190 (1988).

[15] J. Fodor and M. Roubens, *Fuzzy Preference Modelling and Multicriteria Decision Support*, Kluwer, Dordrecht, 1994.

[16] J.M. Łęski and N. Henzel, "ECG baseline wander and power-line interference reduction using nonlinear filter bank", *Signal Processing* 85 (2), 781–793 (2005).

[17] R.R. Yager, "OWA operators in regression problems", *IEEE Trans. Fuzzy Systems* 18 (1), 106–113 (2010).

[18] E.J. Schlossmacher, "An iterative technique for absolute deviations curve fitting", *J. Amer. Statist. Assoc.* 68 (344), 857–859 (1973).

[19] P. Holland and R. Welsch, "Robust regression using iteratively reweighted least-squares", *Commun. Stat. Theoret. Meth.* 6 (9), 813–827 (1977).

[20] D.G. Luenberger, *Linear and Nonlinear Programming*, Kluwer Acad. Press, Boston, 2003.