Linear Regression Models for Interval-Valued Data using Log-transformations

Nykolas Mayko Maia Barbosa*, João Paulo Pordeus Gomes†, César L. C. Mattos†, Diêgo Farias de Oliveira*
Department of Computer Science
Federal University of Ceará
Fortaleza, Brazil
Email: *{nykolas, diegofarias}@lia.ufc.br and †{jpaulo, cesarlincoln}@dc.ufc.br

***Abstract*—Solving linear regression problems on interval-valued data is a challenging task that may arise in many applications. Because of that, many researchers have designed methods for such task in recent years. Although much effort has been devoted to this problem, all available methods rely on modeling the problem as a constrained optimization task, which may lead to sub-optimal results. Moreover, no previous work provide a way to train a model in a incremental way, which is fundamental for big data problems. In this paper, we address both problems by proposing two different linear regression methods based on log-transformations. The proposed methods, referred as Log-transformed OLS for interval data (LOID) and Log-transformed LMS for interval data (LLID), are compared to state-of-the-art methods on both synthetic and real-world datasets. The obtained results indicate the feasibility of our approaches. Furthermore, to the best of our knowledge, LLID is the first sequential linear regression method for interval valued.**

***Keywords*—linear regression, interval data, sequential learning**

I. Introduction

Linear regression models are widely used to predict numerical values that are related to a set of independent measures [1]. To fit the model to the data, it is necessary to estimate a weight vector that defines the linear relationship between the independent variables and the outputs. Such weight vector is usually obtained by minimizing a predefined loss function. Among all possible loss function, the sum of squared errors is the most popular and it defines the so-called least squares regression approach.

Although least squares methods have been successfully used in various applications, all basic formulations are designed to work with datasets where each column represents a variable. In that sense, such models are not suitable to work with interval data. Interval data consists on a representation where each component of a given sample is modeled as an interval, as opposed to a single value. Such data may arise, for instance, due to imprecisions of the measurement devices or data fluctuations in the case of recorded measures during a specific interval of time [2].

Throughout the years many authors have proposed linear regression model for interval valued data. One of the first works is the one by Billard and Diday [3], where they propose a model that considers only the midpoints of the intervals to estimate the regression coefficients. Lima Neto and De Carvalho [4] proposed a method, named Centre and Range Method (CRM), that improves the previous work by considering two independent linear regression problems, where the first models the midpoint and the second models the ranges. In [5], the same authors presented a new version of their method, named Constrained CRM (CCRM), where they use a constrained linear regression to ensure the coherence of the results, i.e., predicted upper bound values shall be higher than predicted lower bounds. The design of sparse Linear models was the object of the work by Giordani [2]. The author proposed a variant of the LASSO algorithm for interval valued data.

By analyzing the aforementioned works, one could notice that the coherence constraint used in both [5] and [2] may impose unnecessary constraints to the final linear models. In [5], the authors assure that the predicted range is always positive by constraining the regression weights to be positive. This rule may lead to sub-optimal results since it does not allow the optimization procedure to find a relation between inputs and outputs given by a negative coefficient, even if it occurs in the data generator. In a different setting, the authors in [2] impose a constraint that makes the range predictions to be positive on the training set, thus not assuring that the predictions on the test set will always be positive. Another drawback of those previous works is that none of the adapted linear regression models provide a way to find the regression coefficients iteratively.

In this work, we aim to tackle both mentioned problems by proposing two linear regression approaches for interval data based on log-transformations of the available data. Roughly speaking, we use the center and range setting and transform the range data using a logarithm function. After that, we compute the coefficients using the OLS (Ordinary Least Squares) or LMS (Least Mean Squares) algorithms. Our method also includes a pre-processing step of standardizing the range data. By standardizing the data before the log-transformation, we may assure that the transformed data is in a quasi-linear region of the logarithm function. This procedure is used to minimize the effect of the nonlinear transformation on the weight estimation procedure. The effectiveness of both proposed methods is verified by comparison to CCRM in several real and synthetic datasets.

The remainder of the paper is organized as follows. Section 2 presents some basic concepts on linear regression models for interval data. In Section 3, we present the proposed algorithms

and its performance is analyzed in Section 4. Final conclusions and future works are discussed in Section 5.

II. Background

A. Basic Concepts of Interval Data

As described in [6], interval data is a class of symbolic data that can be used in order to summarize large datasets. Furthermore, there are many situations in the modern world where the value of a specific variable can be only represented as an interval, such as blood pressure measurement or income information, which is often considered in terms of intervals of the minimum wage. Thus, as emphasized by [7], the availability of this kind of data and the recent growth of machine learning as an important form of data analysis have brought the necessity to extend standard modeling techniques for interval-valued data, an approach sometimes called Symbolic Data Analysis. The importance of this type of data has created another research field that studies and formulates all arithmetic operations on interval variables, as can be seen in [8].

It is relevant to highlight that, following [9], all interval variables used in this work are comprised of closed intervals. We denote an interval variable $X = [x_L,\ x_U]$ as a set of real numbers given by:

$$X = [x_L,\ x_U] = \{x \in \mathcal{R} : x_L\ \leq\ x\ \leq x_U\}.$$

We adopt capital letters to represent an interval variable. Its *endpoints* or *bounds* x_L and x_U are represented by lower case letters, since they are scalar numbers.

We can also define the *range* and *center* of an interval:

- The *range* or *width* of an interval X is given by:

$$x_R = \frac{1}{2}(x_U - x_L).$$

- The *center* or *midpoint* of X is given by:

$$x_C = \frac{1}{2}\ (x_L + x_U).$$

Given the above definitions, we can also represent an interval by using its center and range values:

$$X = [x_C - x_R,\ x_C + x_R].$$

B. Linear Regression for Interval Data

Regression analysis is a technique that can be used to predict the values of a dependent or output quantitative variable as a function of the values of independent or input quantitative variables [10]. The goal of a linear regression task is to figure out a model that fits the data. Thus, it is necessary to estimate a vector of parameters $\hat{\mathbf{w}}$ using the output data vector $\mathbf{y}$ and the input matrix $\mathbf{X}$. The parameter estimation step can be seen as an optimization problem. In our case, we consider standard least squares and gradient descent methods to solve such optimization.We refer the reader to the work in [11], one of the first linear regression methods presented for interval valued-data.

1) The center method: One first approach to estimate $\hat{\mathbf{w}}$ consists in considering only the center of the intervals. We name this method **CM** (Center Method). Given an input matrix $\mathbf{X}^{mn}$ containing m samples with n features each one, and an output vector $\mathbf{y}^m$, the **CM** estimates the parameters as described below according was written in [5]:

$$\hat{\mathbf{w}} = (\mathbf{X}_C^T\ \mathbf{X}_C)^{-1}\ \mathbf{X}_C^T\ \mathbf{y}_C$$

So, given a new vector example $\mathbf{X}^{1n}$, where $\mathbf{X}^{1j} = [x_L^n,\ x_U^n]$, $j \in (1, \cdots, n)$, the value $Y = [y_L,\ y_U]$ will be predicted by $\hat{Y} = [\hat{y}_L,\ \hat{y}_U]$ as follows:

$$\hat{Y}_L = \mathbf{X}_L^{1n}\ \hat{\mathbf{w}} \text{ and } \hat{Y}_U = \mathbf{X}_U^{1n}\ \hat{\mathbf{w}}.$$

As we can see, this method does not ensure that $Y_L \leq Y_U$. If that is not guaranteed, an important rule of interval data will be broken and the result can not be considered a valid interval anymore.

2) The constrained center and range method (CCRM): The method called **CCRM**, proposed in [5], considers inequality constraints in the vector of parameters in order to mathematically ensure that the values of $\hat{Y}_L^i$, where $i \in (1, 2, \ldots, m)$, will be always less than or equal to the values of $\hat{Y}_U^i$.

The core idea consists in applying constraints only over the parameters $\hat{\mathbf{w}}_R$, which ensures that the estimated values $\hat{y}_R^i$ will always be greater than or equal to zero, which also implies that $\hat{y}_L^i$ will always be less than or equal to $\hat{y}_U^i$.

Let us consider $\mathbf{y}_C$ and $\mathbf{y}_R$ as output variables and $\mathbf{x}_C^j$ and $\mathbf{x}_R^j$, $(j\ =\ 1, 2, \ldots, n)$, as input variables related according to the relationship below:

$$y_C^i\ =\ w_C^0 + w_C^1 x_C^{i1} + \cdots + w_C^n x_C^{1n} + \epsilon_C^i,$$

$$y_R^i\ =\ w_R^0 + w_R^1 x_R^{i1} + \cdots + w_R^n x_R^{1n} + \epsilon_R^i,$$

with constraints $w_R^j\ \geq\ 0, j\ =\ 0, \ldots, n$.

Thus, the **CCRM** uses the Least Squares algorithm to predict the $\hat{\mathbf{w}}_C$ and the Lawson and Hanson algorithm [12] adapted to predict the parameters $\hat{\mathbf{w}}_R$.

Following the main idea of **CCRM**, we have developed two new approaches to handle interval-valued data. The first one is named the Log-transformed OLS for Interval Data (**LOID**). The second one is the Log-transformed LMS for Interval Data (**LLID**). Both adapt the data to ensure that $\mathbf{y}_R \geq 0$ and $\mathbf{y}_L \leq \mathbf{y}_U$, using, respectively, Least Squares and Gradient Descent as optimization methods. In the next section both algorithms will be detailed.

III. Proposed Methods

A. Log-transformed OLS for interval data (LOID)

Given $\mathbf{X}^{mn}$ and $\mathbf{y}^m$ as input matrix and output vector, respectively, where $X^{ij}\ =\ [x_L^{ij}, x_U^{ij}]$, $Y_i\ =\ [y_L^i, y_U^i]$, $i\ =\ (1, 2, \cdots, m)$, and $j\ =\ (1, 2, \ldots, n)$. Let us consider $\mathbf{X}_C^{mn}$ and $\mathbf{X}_R^{mn}$ as the centers and ranges of $\mathbf{X}$, respectively. Let also $\mathbf{y}_C^m$ be the centers of $\mathbf{y}^m$ and $\mathbf{y}_R^m$ its ranges. In matrix notation, the proposed **LOID** method can be written as

$$\begin{aligned}\hat{\mathbf{y}}_C &= \mathbf{X}_C\ \hat{\mathbf{w}}_C + \epsilon_C,\\ \hat{\mathbf{y}}_R &= \mathbf{X}_R\ \hat{\mathbf{w}}_R + \epsilon_R,\end{aligned} \tag{1}$$

where, $\hat{\mathbf{w}}_C = (\hat{w}_C^0, \hat{w}_C^1, \ldots, \hat{w}_C^n)$, $\hat{\mathbf{w}}_R = (\hat{w}_R^0, \hat{w}_R^1, \ldots, \hat{w}_R^n)$, $\epsilon_C = (\epsilon_C^1, \ldots, \epsilon_C^n)$, and $\epsilon_R = (\epsilon_R^1, \ldots, \epsilon_R^n)$.

Thus, the sum of squares of errors is given by

$$\begin{aligned}\varepsilon_{LOID} &= \sum_{i=1}^{m}(\epsilon_C^i)^2 + \sum_{i=1}^{m}(\epsilon_R^i)^2\\ &= \sum_{i=1}^{m}(y_C^i - \hat{w}_C^0 - \hat{w}_C^1 x_C^{i1} - \cdots - \hat{w}_C^n x_C^{in})^2\\ &+ \sum_{i=1}^{m}(y_R^i - \hat{w}_R^0 - \hat{w}_R^1 x_R^{i1} - \cdots - \hat{w}_R^n x_R^{in})^2.\end{aligned} \tag{2}$$

The parameters $\hat{\mathbf{w}}_C$ in the first expression of Eq. (1) are given found via standard least squares:

$$\hat{\mathbf{w}}_C = (\mathbf{X}_C^T\ \mathbf{X}_C)^{-1}\ \mathbf{X}_C^T\ \mathbf{y}_C, \tag{3}$$

where we emphasize that the parameters $\hat{\mathbf{w}}$ are not constrained.

The second expression in Eq. (1) must be fitted following the constraint $\mathbf{y}_R \geq 0$. Thus, we ensure the predicted values for $\mathbf{y}_R$ is greater than or equal to zero, using a logarithmic transformation in the training targets $\mathbf{y}_R$, which are used to fit the parameters $\hat{\mathbf{w}}_R$. The logarithmic transformation gives us the possibility to get values from any component of $\hat{\mathbf{w}}_R \leq 0$, an useful feature if we have a dataset that was generated by a function with such behavior, however, applying this tranformantion, we are adding a non-linearity over the data, although there is a way to minimize it, just using a simple standardization in both $\mathbf{X}$ and $\mathbf{y}$, we divided all elements by its maximum value, respectively. Algorithm 1 summarizes the training step of the proposed **LOID** approach.

Algorithm 1 Training step of **LOID**

Require: $\mathbf{X}$, $\mathbf{y}$ {training data set}
1: $\mathbf{X} \leftarrow \dfrac{\mathbf{X}}{max(\mathbf{X})}$ {apply the standardization in $\mathbf{X}$}
2: $\mathbf{y} \leftarrow \dfrac{\mathbf{y}}{max(\mathbf{y})}$ {apply the standardization in $\mathbf{y}$}
3: $m, n \leftarrow dim(\mathbf{X})$ {get the dimension of input matrix}
4: $\mathbf{y}_{train} \leftarrow \log \mathbf{y}$ {apply the log-transformation in outputs samples}
5: $\mathbf{X}_{train} \leftarrow [\mathbf{1}^{m1}\ \mathbf{X}]_{m(n+1)}$ {add the bias}
6: $\hat{\mathbf{w}} \leftarrow (\mathbf{X}_{train}^T \mathbf{X}_{train})^{-1} \mathbf{X}_{train}^T \mathbf{y}_{train}$

Given a new example $\mathbf{z}_{1,n}$, where $Z^j = [z_L^j, z_U^j]$, $j = (1, \cdots, n)$ we can predict $\hat{y}$ according the steps summarized in Algorithm 2. It is import to highlight that, as can be seen in lines 3 and 4, first, we predict an output transformed by the logarithm function applied in Alg. 1, but, since our goal is to estimate the outcome from an input without the such transformation, we must apply the inverse function of the logarithm, that is, the exponential function.

Algorithm 2 Testing step of **LOID**

Require: $\mathbf{X}$ {test data set}
1: $m, n \leftarrow dim(\mathbf{X})$ {get the dimension of input matrix}
2: $\mathbf{X}_{train} \leftarrow [\mathbf{1}^{m1}\ \mathbf{X}]_{m(n+1)}$ {add the bias}
3: $\hat{\mathbf{y}}_{aux} \leftarrow \mathbf{X}_{test}\ \hat{\mathbf{w}}^T$ {predict the output transformed}
4: $\hat{\mathbf{y}} \leftarrow \exp(\hat{\mathbf{y}}_{aux})$ {predict the output}

B. Log-transformed LMS for interval data (LLID)

In this section we present an extension to interval-valued data for the highly popular least-mean-squares (LMS) algorithm, which was developed by [13] to enable iterative learning with linear models.

Let $\mathbf{X}^{mn}$ and $\mathbf{y}^m$ be an input matrix and an output vector, respectively, where each X^{ij} is represented by an interval feature $X_{ij} = [x_L^{ij}, x_U^{ij}]$, $i = (1, \cdots, m)$, and $j = (1, \cdots, n)$. Let also Y_i be an interval representing each correspondent output $Y_i = [y_L^i, y_U^i]$. Let us consider $\mathbf{X}_C^{mn}$ and $\mathbf{X}_R^{mn}$ the centers and ranges from $\mathbf{X}$, respectively. We also have $\mathbf{y}_C^m$ as the centers of $\mathbf{y}^m$ and $\mathbf{y}_R^m$ as its ranges.

Assuming the existence of a continuous mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$ between the input and the output space, our goal is to estimate f from the available data. We consider the function f to be linear, i.e:

$$\hat{y}_i = f(\mathbf{x}^i) = \mathbf{x}^i \mathbf{w}. \tag{4}$$

Thus, following to the presentation of the **LOID** method in the latter section, our proposed extension for the LMS, named **LLID**, aims to fit two different functions, one based on $\mathbf{w}_C$ and other based on $\mathbf{w}_R$. The latter must consider the constraint $\hat{\mathbf{y}}_R \geq 0$. Thus, we can rewrite the Eq. (4) twice as follows:

$$\begin{aligned}\hat{y}_C^i &= f(\mathbf{x}_C^i) = \mathbf{x}_C^i\ \hat{\mathbf{w}}_C,\\ \hat{y}_R^i &= f(\mathbf{x}_R^i) = \mathbf{x}_R^i\ \hat{\mathbf{w}}_R.\end{aligned} \tag{5}$$

We can compute the output error in the i-th sample:

$$\begin{aligned}\epsilon_C^i &= y_C^i - \hat{y}_C^i,\\ \epsilon_R^i &= \log(y_R^i) - \hat{y}_R^i.\end{aligned} \tag{6}$$

Substituting Eq. (5) into these expressions yields

$$\begin{aligned}\epsilon_C^i &= y_C^i - \mathbf{x}_C^i \hat{\mathbf{w}}_C,\\ \epsilon_R^i &= y_R^i - \mathbf{x}_R^i \hat{\mathbf{w}}_R.\end{aligned}$$

Consider cost functions $J(\hat{\mathbf{w}}_C) = \frac{1}{2}\sum_{i=1}^{m} \epsilon_C^i$ and $J(\hat{\mathbf{w}}_R) = \frac{1}{2}\sum_{i=1}^{m} \epsilon_R^i$, which are both continuously differentiable functions of some unknown weights (parameters) vectors $\hat{\mathbf{w}}_C$ and $\hat{\mathbf{w}}_R$. The function $J(\hat{\mathbf{w}})$ maps the elements of $\hat{\mathbf{w}}$ into real numbers. We want to find an optimal solution $\hat{\mathbf{w}}^*$, i.e., $\hat{\mathbf{w}}_C^*$ and $\hat{\mathbf{w}}_R^*$, that satisfies the condition $J(\hat{\mathbf{w}}^*) \leq J(\hat{\mathbf{w}})$.

The solution can be obtained by solving the optimization problem below:

$$\begin{aligned}&\underset{\mathbf{w}_C}{\text{minimize}}\ J(\hat{\mathbf{w}}_C),\\ &\underset{\mathbf{w}_R}{\text{minimize}}\ J(\hat{\mathbf{w}}_R).\end{aligned} \tag{7}$$

Algorithm 3 Training step of **LLID**

Require: $\mathbf{X}$, $\mathbf{y}$, $epochs$, α {training dataset, number of epochs and learning rate}

1: $\mathbf{X} \leftarrow \frac{\mathbf{X}}{max(\mathbf{X})}$ {apply the standardization in $\mathbf{X}$}
2: $\mathbf{y} \leftarrow \frac{\mathbf{y}}{max(\mathbf{y})}$ {apply the standardization in $\mathbf{y}$}
3: $m, n \leftarrow dim(\mathbf{X})$ {get the dimension of input matrix}
4: $\mathbf{X}_{train} \leftarrow [\mathbf{1}^{m1}\ \mathbf{X}]_{m\ \mathrm{x}\ (n+1)}$
5: $\mathbf{y}_{train} \leftarrow \log \mathbf{y}$ {apply the log-transformation in outputs samples}
6: $\hat{\mathbf{w}}^{1(n+1)} \leftarrow [\overline{\mathbf{y}}_{train}\ \mathbf{0}^{1\ \mathrm{x}\ n}]$
7: **for** $e \in range(epochs)$ **do**
8: $\quad indices \leftarrow permutation(m)$
9: $\quad$ **for** $i \in indices$ **do**
10: $\quad\quad \hat{\mathbf{y}} \leftarrow \mathbf{x}^i_{train}\ \hat{\mathbf{w}}^T$
11: $\quad\quad \epsilon \leftarrow \mathbf{y}^i_{train} - \hat{\mathbf{y}}$
12: $\quad\quad \hat{\mathbf{w}} \leftarrow \hat{\mathbf{w}} + \alpha\ \epsilon\ \mathbf{x}^i_{train}$
13: $\quad$ **end for**
14: **end for**

Following a stochastic gradient optimization approach, it is easy to show that in general the weights can be updated as follows [14], [15]:

$$\begin{aligned} \hat{w}^0 &= 0 \\ \epsilon^i &= y^i - \hat{w}^{i-1}\mathbf{x}^i, \\ \hat{w}^i &= \hat{w}^{i-1} + \alpha\ \epsilon^i\ \mathbf{x}^i, \end{aligned}$$

where ϵ is computed using the expressions in Eq. (6) and α is the learning rate.

Algorithm 3 summarizes the above operations. Note that the sixth line explains how we initialize the first component of $\hat{\mathbf{w}}$, the *bias*, with the mean value of $\mathbf{y}_{train} = \log(\mathbf{y})$.

Similar to the **LOID** algorithm detailed in the latter section, after the training step we may proceed to the test step following Alg. 4, which must consider the application of the *exponential* function in the model predictions.

Algorithm 4 Testing step of **LLID**

Require: $\mathbf{X}$, $\mathbf{y}$ {test data set}

1: $m, n \leftarrow dim(\mathbf{X})$ {get the dimension of input matrix}
2: $\mathbf{X_{train}} \leftarrow [\mathbf{1}^{m1}\ \mathbf{X}]_{m\ \mathrm{x}\ (n+1)}$ {add the bias}
3: $\hat{\mathbf{y}}_{aux} \leftarrow \mathbf{X}_{test}\ \hat{\mathbf{w}}^T$ {predict the output transformed}
4: $\hat{\mathbf{y}} \leftarrow \exp(\hat{\mathbf{y}}_{aux})$ {predict the output}

Having described our two approaches, in the next section we will explain and discuss the experimental results performed in several tests scenarios using both real and synthetic datasets.

IV. Experimental Results

First, the usefulness of the linear methods proposed in this paper will be evaluated via experiments with synthetic interval-valued datasets with different linear configurations. Initially, we evaluate the performance of the optimization algorithms (**CCRM**, **LLID**, **LLOID**) to identify which algorithm presents the best accuracy for the parameter estimation.

After that, the prediction performance of these methods will be compared in terms of the root-mean-square error of the center ($RMSE_C$), range ($RMSE_R$), and summed lower and upper ($RMSE_{LU}$) values for each interval. We follow an experimental Monte Carlo scheme with 50 repetitions. The *RMSE* functions are computed as:

$$\begin{aligned} RMSE_C &= \sqrt{\frac{\sum_{i=i}^{m}(y^i_C - \hat{y}^i_C)}{m}}, \\ RMSE_R &= \sqrt{\frac{\sum_{i=i}^{m}(y^i_R - \hat{y}^i_R)}{m}}, \\ RMSE_{LU} &= \sqrt{\frac{\sum_{i=i}^{m}(y^i_L - \hat{y}^i_L)}{m}} + \sqrt{\frac{\sum_{i=i}^{m}(y^i_U - \hat{y}^i_U)}{m}}, \end{aligned} \tag{8}$$

where m is the number of observations on the test step.

We have chosen to use the above metrics to quantify the errors in each separated model, one to predict the $\mathbf{y}_R$ and another to predict $\mathbf{y}_C$. Thus, by analyzing the $RMSE_R$ and $RMSE_C$ values we can compute the accuracy of the evaluated methods in each test scenario. On the other hand, we are also interested in the overall interval error achieved by each technique. So, we also compute the $RMSE_{LU}$ that quantifies the error by computing the sum of the lower and upper bounds' errors.

Finally, all the models will also be evaluated using real interval-valued datasets. Their performances will be compared considering a Leave One-Out (LOO) validation scheme. LOO was used because the chosen real datasets are smaller than the synthetic ones. In average, we have around 30 observations in each set.

A. Synthetic Datasets

The synthetic datasets were generated using 2 strategies. The first one considers the configuration presented in [5], as will be soon described. Furthermore, we have created 2 addtional synthetic datasets. In all cases, we separated 67% of the generated data for the training step and 33% for the test step.

1) Synthetic Datasets used in [5]: As presented in Tab. I, we chose two distinct configurations proposed in [5]. The configuration called C has the center and range independents, while the D datasets consider a dependence between midpoint and range. Both configurations, denoted by C_p and D_p, were generated with dimension p equal to 1 and 3, respectively.

The construction of the C standard datasets and the corresponding interval data sets is carried out in the following steps:

1) Generate $\mathbf{X}_C$, $\mathbf{X}_R$, and $\mathbf{y}_C$ according to the Tab. I;
2) Compute the random variable $\mathbf{y}_C$:

$$\mathbf{y}_C = (\mathbf{X}_C)^T\ \boldsymbol{w} + \boldsymbol{\epsilon},$$

where $(\mathbf{X}_C)^T = (1,\ X^1_C)$ and $\boldsymbol{w} \sim U[c,\ d]$ in $\mathcal{R}^2$, or $(\mathbf{X}_C)^T = (1,\ X^1_C, X^2_C, X^3_C)$ and $\boldsymbol{w} \sim U[c,\ d]$ in $\mathcal{R}^4$ and $\boldsymbol{\epsilon} \sim U[e,\ f]$.

TABLE I
CONFIGURATIONS OF SYNTHETIC DATASETS PROPOSED BY [5].

C_1	$x_C^j \sim U[20,\ 40]$	$x_R^j \sim U[20,\ 40]$	$y_R^j \sim U[20,\ 40]$	$\epsilon \sim U[-20,\ 20]$
C_3	$x_C^j \sim U[20,\ 40]$	$x_R^j \sim U[1,\ 5]$	$y_R^j \sim U[1,\ 5]$	$\epsilon \sim U[-20,\ 20]$
D_1	$x_C^j \sim U[20,\ 40]$	$\epsilon \sim U[-20,\ 20]$	$\epsilon^* \sim U[1,\ 5]$	
D_3	$x_C^j \sim U[20,\ 40]$	$\epsilon \sim U[-5,\ 5]$	$\epsilon^* \sim U[1,\ 5]$	

TABLE II
DESCRIPTION OF REAL DATASETS

	Number of Observations	Number of Features
Cardiologic	59	2
Car	33	2
Mushroom	23	2
Soccer	20	2
Nasa	13	2

where U is a uniform distribution.

The generation of the D standard datasets is described below:

1) Generate $\mathbf{X}_C$ according to the Tab. I;
2) Compute the random variable $\mathbf{y}_C$: $\mathbf{y}_C = (\mathbf{X}_C)^T\ \boldsymbol{w} + \boldsymbol{\epsilon}$, where $(\mathbf{X}_C)^T = (1,\ X_C^1)$ and $\boldsymbol{w} \sim U[c,\ d]$) in $\mathcal{R}^2$, or $(\mathbf{X}_C)^T = (1,\ X_C^1, X_C^2, X_C^3)$ and $\boldsymbol{w} \sim U[c,\ d]$ in $\mathcal{R}^4$ and $\boldsymbol{\epsilon} \sim U[e,\ f]$;
3) Compute the random variable $\mathbf{y}_R$: $\mathbf{y}_R = \mathbf{y}_C\ \boldsymbol{w^*} + \boldsymbol{\epsilon}^*$, where the variables X_R^j ($j\ =\ 1, \cdots, n$) are related to variables X_C^j according to $X_R^j = X_C^j\ \boldsymbol{w}^* + \boldsymbol{\epsilon}^*$, where $\boldsymbol{w}^* \sim U[g,\ h]$ and $\boldsymbol{\epsilon}^* \sim U[i,\ j]$.

2) Proposed Synthetic Datasets: The **CCRM** method constrains the predicted parameters $\hat{\mathbf{w}}_R$ to be greater than or equal to zero. In this paper, we have generated 2 datasets, 1-dimensional (A_1) and 3-dimensional (A_3), where the true parameters $\mathbf{w}_R$ are less than zero. Below, we detail the functions that generated these datasets.

The dataset A_1 was generated following those steps:

1) Compute $\mathbf{X}_C = (\mathbf{1},\ U[20,\ 40])$, $\mathbf{X}_R = (\mathbf{1},\ \ U[0,\ 1])$;
2) Compute $\mathbf{y}_C = \mathbf{X}_C\ *\ (2,\ 5)^T$ and $\mathbf{y}_R = \mathbf{X}_R(5,\ -3)^T$;
3) Compute the variance of $S_y^C = \mathbf{y}_C$ and $S_y^R = \mathbf{y}_R$;
4) Recompute $\mathbf{y}_C = \mathbf{y}_C + \sqrt{0.01 S_y^C}\ \boldsymbol{\epsilon}$ and $\mathbf{y}_R = \mathbf{y}_R + \sqrt{0.01 S_y^R}\ \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
5) For values $y_R^i < 0$, replace them with $y_R^i = 0$.

The dataset A_3 was generated as follows:

1) Compute $\mathbf{X}_C = (\mathbf{1},\ U[20,\ 40])$, $\mathbf{X}_R = (\mathbf{1},\ U[1,\ 10])$;
2) Compute $\mathbf{y}_C\ =\ \mathbf{X}_C(2,\ \ 2,\ \ 5,\ \ 6)^T$ and $\mathbf{y}_R\ =\ \mathbf{X}_R(62,\ -2,\ -3,\ -1)^T$;
3) Compute the variance of $S_y^C = \mathbf{y}_C$ and $S_y^R = \mathbf{y}_R$;
4) Recompute $\mathbf{y}_C = \mathbf{y}_C + \sqrt{0.01 S_y^C}\ \boldsymbol{\epsilon}$ and $\mathbf{y}_R = \mathbf{y}_R + \sqrt{0.01 S_y^R}\ \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
5) For values $y_R^i < 0$, replace them with $y_R^i = 0$.

B. Real-world Datasets

We conducted experiments with five real-world datasets that can be found in [16]. A brief description of each dataset is presented in Tab. II.

C. Discussion

The results of the synthetic datasets are presented in Tab. III. As can be noticed, the proposed variants of LMS and OLS achieved the good results in the synthetic datasets proposed in CCRM's paper even the **CCRM** have gotten best result, our proposed method is really close, on the other hand, in the datasets proposed by this paper, A_1 and A_3, which have the weights of the generation function less than zero, **LLID** and **LOID** achived the best results, while **CCRM** did not get good results, because its model is constrains the weights doing the $\hat{\mathbf{w}} \approx 0$, which increases the error substantially.

The results obtained with real-world datasets are summarized in Tab. IV. In general, except for the Nasa dataset, where the **LOID** result was worse in comparison with the other methods, for all the datasets, the proposed methods were closer or better than **CCRM**. Overall, the best results were achieved by **LLID**.

V. CONCLUSION

In this paper, we proposed two linear regression models for interval-valued data. The methods, named Log-transformed OLS for interval data (LOID) and Log-transformed LMS for interval data (LLID), comprise a standardization step of the input and output data followed by a log-tansformation of the output data. After that, OLS or LMS methods can be used to estimate the model coefficients.

Our proposals have two advantages over its most popular counterparts: i) both LOID and LLID do not require any constraint on the optimization and ii) LLID is the first sequential linear regression method for interval valued. That fact makes LLID suitable for big data applications.

Based on an extensive the set of experiments, we could verify that LOID and LLID outperformed CCRM is various situations. Hence, we believe that both proposals can be viewed as valid alternative to tackle the linear regression of interval data problem.

ACKNOWLEDGMENTS

This work was supported by CAPES and Federal University of Ceará. The authors would also like to thank the Brazilian National Council for Scientific and Technological Development (CNPq), ASTEF (grants 305048/2016-3 and F0191

TABLE III
COMPARISON BETWEEN **LOID**, **LLID** AND **CCRM** IN SYNTHETICS DATASETS USING *RMSE* AS THE METRIC OF ERROR COMPUTATION.

		A_1	C_1	D_1	A_3	C_3	D_3
LOID	$RMSE_C$	$\mathbf{3.12 \pm 1.87e{-1}}$	$1.19e1 \pm 3.77e{-1}$	$2.80 \pm 9.80e{-2}$	$\mathbf{5.02 \pm 3.62e{-1}}$	$1.19e1 \pm 4.05e{-1}$	$2.90 \pm 9.80e{-2}$
	$RMSE_R$	$\mathbf{1.23e{-1} \pm 9.00e{-3}}$	$1.23 \pm 4.20e{-2}$	$1.24e1 \pm 7.76e{-1}$	$\mathbf{2.52 \pm 5.81e{-1}}$	$1.18e1 \pm 4.20e{-2}$	$1.48e1 \pm 7.33e{-1}$
	$RMSE_{LU}$	$\mathbf{6.24 \pm 3.75e{-1}}$	$2.40e1 \pm 7.50e{-1}$	$2.50e1 \pm 1.15$	$\mathbf{1.12e1 \pm 8.46e{-1}}$	$2.39e1 \pm 8.05e{-1}$	$2.98e1 \pm 1.45$
LLID	$RMSE_C$	$\mathbf{3.12 \pm 1.87e{-1}}$	$1.19e1 \pm 3.77e{-1}$	$2.80 \pm 9.80e{-2}$	$\mathbf{5.02 \pm 3.62e{-1}}$	$1.19e1 \pm 4.05e{-1}$	$2.90 \pm 9.80e{-2}$
	$RMSE_R$	$\mathbf{1.23e{-1} \pm 9.00e{-3}}$	$1.23 \pm 4.20e{-2}$	$1.24e1 \pm 7.76e{-1}$	$\mathbf{2.52 \pm 5.81e{-1}}$	$1.18e1 \pm 4.20e{-2}$	$1.48e1 \pm 7.34e{-1}$
	$RMSE_{LU}$	$\mathbf{6.24 \pm 3.75e{-1}}$	$2.40e1 \pm 7.50e{-1}$	$2.50e1 \pm 1.15$	$\mathbf{1.12e1 \pm 8.46e{-1}}$	$2.39e1 \pm 8.05e{-1}$	$2.98e1 \pm 1.45$
CCRM	$RMSE_C$	$3.12 \pm 1.87e{-1}$	$\mathbf{1.19e1 \pm 3.77e{-1}}$	$\mathbf{2.80 \pm 9.80e{-2}}$	$5.02 \pm 3.62e{-1}$	$1.19e1 \pm 4.05e{-1}$	$\mathbf{2.90 \pm 9.80e{-2}}$
	$RMSE_R$	$8.57e{-1} \pm 4.00e{-2}$	$\mathbf{1.20 \pm 3.9e{-2}}$	$\mathbf{1.17e1 \pm 4.68e{-1}}$	$9.21 \pm 5.91e{-1}$	$1.15e1 \pm 3.70e{-2}$	$\mathbf{1.32e1 \pm 6.40e{-1}}$
	$RMSE_{LU}$	$6.46 \pm 3.63e{-1}$	$\mathbf{2.40e1 \pm 7.50e{-1}}$	$\mathbf{2.34e1 \pm 9.37e{-1}}$	$2.09e1 \pm 1.03$	$2.39e1 \pm 8.05e{-1}$	$\mathbf{2.66e1 \pm 1.26}$

TABLE IV
COMPARISON BETWEEN **LOID**, **LLID** AND **CCRM** IN REALS DATASETS USING *RMSE* AS THE METRIC OF ERROR COMPUTATION.

		Cardiologic	**Car**	**Mushroom**	**Soccer**	**Nasa**
LOID	$RMSE_C$	8.96 ± 6.92	$2.32e4 \pm 1.60e4$	1.62 ± 1.27	1.84 ± 1.49	$4.92e2 \pm 4.52e2$
	$RMSE_R$	5.68 ± 4.90	$1.51e4 \pm 2.38e4$	$1.05 \pm 7.68e{-1}$	$1.10 \pm 7.74e{-1}$	$5.44e2 \pm 7.43e2$
	$RMSE_{LU}$	$2.12e1 \pm 1.31e1$	$5.89e4 \pm 4.45e4$	3.83 ± 2.30	3.95 ± 2.86	$1.26e3 \pm 1.40e3$
LLID	$RMSE_C$	$\mathbf{8.86 \pm 6.80}$	$\mathbf{2.20e4 \pm 1.80e4}$	1.65 ± 1.07	1.89 ± 1.59	$\mathbf{3.79e2 \pm 3.58e2}$
	$RMSE_R$	$\mathbf{5.70 \pm 4.90}$	$\mathbf{1.48e4 \pm 2.34e4}$	$1.06 \pm 8.39e{-1}$	$1.22 \pm 9.43e{-1}$	$\mathbf{3.56e2 \pm 4.79e2}$
	$RMSE_{LU}$	$\mathbf{2.10e1 \pm 1.3e1}$	$\mathbf{5.69e4 \pm 4.68e4}$	3.78 ± 2.06	4.29 ± 3.01	$\mathbf{9.14e2 \pm 8.73e2}$
CCRM	$RMSE_C$	8.96 ± 6.92	$2.32e4 \pm 1.60e4$	$\mathbf{1.62 \pm 1.27}$	$\mathbf{1.84 \pm 1.49}$	$4.92e2 \pm 4.53e2$
	$RMSE_R$	5.69 ± 4.68	$1.98e4 \pm 2.17e4$	$\mathbf{9.27e{-1} \pm 6.79e{-1}}$	$\mathbf{1.07 \pm 8.19e{-1}}$	$3.97e2 \pm 3.64e2$
	$RMSE_{LU}$	$2.12e1 \pm 1.29e1$	$6.14e4 \pm 4.21e4$	$\mathbf{3.52 \pm 2.40}$	$\mathbf{3.93 \pm 2.87}$	$9.84e2 \pm 9.06e2$

respectively) and Foundation of Scientific and Technological Development in Ceará for the financial support.

REFERENCES

[1] E. d. A. Lima Neto and F. d. A. T. de Carvalho, "Nonlinear regression applied to interval-valued data," *Pattern Analysis and Applications*, vol. 20, no. 3, pp. 809–824, Aug 2017. [Online]. Available: https://doi.org/10.1007/s10044-016-0538-y

[2] P. Giordani, "Regression analysis for interval-valued data based on the lasso technique," *Technical Report n. 7, Department of Statistical Sciences, Sapienza University of Rome*, 2011.

[3] L. Billard and E. Diday, "Regression analysis for interval-valued data," in *Data Analysis, Classification, and Related Methods*, H. A. L. Kiers, J.-P. Rasson, P. J. F. Groenen, and M. Schader, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 369–374.

[4] E. de A. Lima Neto and F. de A.T. de Carvalho, "Centre and range method for fitting a linear regression model to symbolic interval data," *Computational Statistics & Data Analysis*, vol. 52, no. 3, pp. 1500–1515, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167947307001934

[5] E. A. L. Neto and F. A. Carvalho, "Constrained linear regression models for symbolic interval-valued variables," *Computational Statistics & Data Analysis*, vol. 54, no. 2, pp. 333–347, 2010.

[6] H.-H. Bock and E. Diday, *Analysis of symbolic data: exploratory methods for extracting statistical information from complex data*. Springer Science & Business Media, 2012.

[7] E. d. A. L. Neto and F. d. A. de Carvalho, "Nonlinear regression applied to interval-valued data," *Pattern Analysis and Applications*, vol. 20, no. 3, pp. 809–824, 2017.

[8] R. B. Kearfott, "Interval computations: Introduction, uses, and resources," *Euromath Bulletin*, vol. 2, no. 1, pp. 95–112, 1996.

[9] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to interval analysis*. Siam, 2009, vol. 110.

[10] D. G. Kleinbaum, L. L. Kupper, K. E. Muller, and A. Nizam, *Applied regression analysis and other multivariable methods*. Duxbury Press Belmont, CA, 1988, vol. 601.

[11] L. Billard and E. Diday, "Regression analysis for interval-valued data," in *Data Analysis, Classification, and Related Methods*. Springer, 2000, pp. 369–374.

[12] C. L. Lawson and R. J. Hanson, *Solving least squares problems*. Siam, 1995, vol. 15.

[13] B. Widrow and M. E. Hoff, "Adaptive switching circuits," Stanford Univ Ca Stanford Electronics Labs, Tech. Rep., 1960.

[14] S. S. Haykin, *Adaptive filter theory*. Pearson Education India, 2005.

[15] S. S. Haykin *et al.*, *Neural networks and learning machines/Simon Haykin.* New York: Prentice Hall,, 2009.

[16] R. A. A. Fagundes, "Métodos de regressão robusta e kernel para dados intervalares," Ph.D. dissertation, Universidade Federal de Pernambuco, 2013.