Robust weighted regression via PAELLA sample weights

Manuel Castejón-Limas[a], Hector Alaiz-Moreton[b], Laura Fernández-Robles[a], Javier Alfonso-Cendón[a], Camino Fernández-Llamas[a], Lidia Sánchez-González[a], Hilde Pérez[a]

[a]*Department of Mechanical, Computer Science and Aerospace Engineering, Universidad de León, 24071, León, Spain*
[b]*Department of Electrical, Systems and Automatic Engineering, Universidad de León, 24071, León, Spain*

Abstract

This paper reports the usage of the occurrence vector provided by the PAELLA algorithm in the context of robust regression. PAELLA was originally conceived as an outlier detection and data cleaning technique. A novel approach is to use this algorithm not for discarding outliers but to generate information related to the reliability of the observations recorded in the dataset. This approach proves to provide successful results when compared to traditional common practice such as outlier removal. A set of experiments using a contrived difficult artificial dataset are described using both neural networks and classical polynomial fitting. Finally, a successful comparison of our approach to two state-of-the-art algorithms proves the benefits of using

*manuel.castejon@unileon.es
Email addresses: manuel.castejon@unileon.es (Manuel Castejón-Limas), hector.moreton@unileon.es (Hector Alaiz-Moreton), l.fernandez@unileon.es (Laura Fernández-Robles), javier.alfonso@unileon.es (Javier Alfonso-Cendón), camino.fernandez@unileon.es (Camino Fernández-Llamas), lidia.sanchez@unileon.es (Lidia Sánchez-González), hilde.perez@unileon.es (Hilde Pérez)

the PAELLA algorithm in the context of robust regression.

Keywords: weighted regression, robust regression, outlier detection, PAELLA, multilayer perceptron

---

1. Introduction

Experimental sciences rely on data to progress. Either to prove hypotheses or to build new models, datasets are the cornerstone on which to build the foundations of modern science. During centuries, now considered small datasets supported the development of scientific descriptions of the observed phenomena. Such datasets were carefully handcrafted by scientist in experiments under controlled conditions. In such scenario, perturbations, noise, and outliers had a limited presence due to the efforts of those designing and running the experiments.

The advent of computers and the availability of huge raw data sources have proven to be game changers. Last decade has witnessed the growth of a subtly different approach in science. The size of the datasets has exploded in many fields and scientists are getting more and more accustomed to analyzing huge datasets of observations from real world processes not constrained to controlled conditions. In other words, scientists are becoming familiar with studying the real processes in the wild instead of designing a laboratory replica.

Such huge datasets from the wild differ substantially from their traditional laboratory counterparts in the reliability of the samples. Leaving out the controlled conditions and the attentive measurement procedures, all that is left is the unknown origin of the samples. In raw datasets the researchers

always have to question on the truthfulness of the data as each observation might have been originated by the process under study, or can be the result of foreign perturbations: an outdoors humidity sensor easily gets saturated for hours by morning dew, turning on high power electrical machines produces spike signals in nearby circuitry, lack of proper calibration and maintenance of thermal cameras produce biased images to process, or just human beings making mistakes while annotating information, are just a few examples of perturbation sources that cause outliers to appear in the datasets [1].

In this new world, researchers strive to handle the complexity added by these unfiltered data sources in fields as different as environmental modeling [2–5], multimedia mobile health applications [6], or factories' process optimization [7–9]. The main strategies for coping with this complexity are filtering the data on a preprocessing stage [10], and applying robust techniques capable of providing satisfactory results no matter whether the data was partially corrupted [11]. An important advantage of the latter is that it takes advantage of all the information collected, which is crucial in the context of not so big datasets.

This paper reports the successful experience of using PAELLA [12], an algorithm that was originally intended for outlier identification and filtering, but that will prove useful as well as a predictive tool in the context of robust regression.

1.1. Experiments that lead to our current proposal

Previous work [13] reported the first attempt to take advantage of the extra information contained in the occurrence vector[1] of PAELLA algorithm [12]. For such purpose, a contrived difficult dataset was tested on a set of different experiments, described below.

First, Castejon et al. [13] reported a reference regression discarding those samples marked as outliers by using PAELLA as an outlier identification technique; that is, using the standard binary vector obtained after thresholding the occurrence vector. This experiments serves the purpose of providing a baseline to measure the benefits obtained by the competing approaches.

The second experiment reported the results from a natural extension of the first experiment. Instead of classifying the samples in two categories by comparing the values in the occurrence vector to an established threshold, multiple categories can be defined just by using several thresholds, slicing the occurrence vector into several bins. For a particular number of bins N, the paper reported the results from fitting a battery of models, each of which used a different training dataset comprising some of the bins in an exhaustive manner so that the 2^N different possibilities were used. It is clear, then, that as the bins get narrower, the computational cost of this approach makes it unfeasible for relatively low number of categories. Interestingly, the conclusion provided by the experiment was that the higher the number of bins, the better the results. Thus, an abstraction of this approach is required

[1]This vector is a result from applying the PAELLA algorithm. It represents the rate in which the observations from the dataset were considered outliers along the number of runs performed by PAELLA.

in order to enlarge the number of bins while keeping computational costs restrained.

The third experiment reported one possible such abstraction. The samples then played the role of the bins, as in the limit the bins would be so narrow as to contain a single sample. The model was fitted with a random sample of the dataset, the sampling likelihood was equal to the corresponding value of the occurrence vector for each observation. The conclusion of this experiment was that results similar to those provided by the baseline experiment could be obtained by following this approach, with the benefit of not discarding any sample out of the experiment.

In what follows we describe another different abstraction that extends the concepts used in that third experiment: using the occurrence vector as an input to sample weighted regression, both in classical regression and in neural networks.

1.2. Structure of the paper

The rest of the paper is structured as follows. In Section 2 the methods used along the paper are described. Section 3 elaborates on the core of the experiments reported in this paper with the aim of comparing the results of using PAELLA boosted weighted regression to common practice and state-of-the-art methods. In Section 3.1 we report the experiments performed using a noisy dataset and three different methods to cope with outliers: outlier removal, macro sampling and weighted regression using linear models. In Sections 3.2 and 3.3 we compare the suggested approach now using multilayer perceptron and linear models to two state-of-the-art methods. Finally, we draw the conclusions of our work in Section 4.

2. Methods

2.1. PAELLA algorithm background

PAELLA was originally conceived as a preprocessor filter for cleaning the raw datasets. In such operating mode, PAELLA was capable of identifying which the observations following a common behavior were, and which showed distinctive features. That is, each observation was labeled as belonging to the common pattern, or as an outlier. Thus, the main results from the PAELLA algorithm could originally be expressed as a binary vector with the resulting classification.

Each of these groups from the PAELLA identification were then subjected to different treatments: the regular samples were used in subsequent stages of analysis such as modeling; while outliers ought to be further analyzed to discover the causes of their occurrence. This kind of exercise provided interesting results: on one hand a predictive model obtained from clean data, thus mitigating the interference of outliers in the estimation of the model's parameters; on the other hand, curated samples from abnormal behaviors that helped in spotting critical parts of the process that were producing undesired consequences.

An interesting feature of PAELLA in the outlier identification arena is that it works with both normal and non-normal multivariate data, which makes it specially useful for real datasets. In order to produce its results, the dataset might first be divided in a given number of groups using a clustering technique. This previous clustering is optional, but in many situations it improves the final results. After this previous clustering, the PAELLA algorithm follows three phases as described as pseudocode in the algorithms

Phase 1, Phase 2, and Phase 3. In Phase 1 the dataset is coated by a set of hypersurfaces similarly to a collection of tiles covering the observations. This phase spots potential outliers as those foreign to the hypersurfaces collection. Phase 2 delivers a list of potential outliers. Phase 3 delivers a list of the samples that were annotated as outliers most often.

Phase 1 Sample coating using a collection of hypersurfaces

Require: A dataset, and optionally a classification vector from a clustering algorithm

Ensure: A collection of hypersurfaces coating the dataset

1: A single sample from each cluster is randomly chosen as a seed point
2: The rest of the samples in a particular subset are classified in accordance to their Mahalanobis distance to the seed point
3: The closest samples are added to a set of samples, G_i, that conform to the model and are used to fit a model
4: The sample points not used for fitting the model are used to calculate their residual and; those with smallest residuals are regarded as compliant with the model and added to the G_i subset.
5: **repeat**
6: steps 1, 2, 3 and 4 using those samples not yet in G_i
7: **until** Time permits

As originally presented, PAELLA delivers a binary classification. Nevertheless, as stated in the algorithm, this binary classification is the transformation of the frequency of being annotated as outlier when compared to a specific threshold. Previous research [13, 14] explored the potential benefits of using the occurrence vector instead of the binary classification. Next

Phase 2 Iterative outlier detection

Require: A collection of hypersurfaces coating the dataset

Ensure: List of potential outliers

1: The observations are confronted versus the collection of models

2: The smallest residuals of the samples are used to associate the samples to the models

3: The samples with the biggest residuals are annotated

Phase 3 Reduce previous iterations

Require: A list of potential outliers in a particular iteration

Ensure: A list of potential outliers

1: **repeat**

2: Phases 1 and 2

3: Reduce the results from previous iterations summarizing in a vector the frequencies of outlierness for all samples. This vector is named occurrence vector v.

4: **until** As time permits

5: Those samples that were annotated as outliers above a frequency threshold t are considered as outliers and separated for further analysis.

subsections present the methods considered in this work based on PAELLA algorithm.

2.2. Outlier removal and neural network regression

This method consists of using the PAELLA algorithm for outlier identification and removal as initially designed in [12]. The samples in the dataset identified as outliers are not taken into account for predicting the model. The training set is split into two categories: outliers and the rest, accordingly to a given value of the frequency threshold.

We train a set of multilayer perceptron (MLP) models with the training samples that are not identified as outliers. The training strategy used for adjusting the parameters is the stochastic gradient descent (SGD) with momentum method [15]. In order to avoid overfitting of the MLP models we use the early stopping method, splitting the training set into training and validation subsets following 66%, 34% proportions, respectively. We choose the set of optimal hyperparameters by means of an exhaustive grid search.

2.3. Outlier removal and linear regression

Similarly as in Section 2.2, the outliers are identified and set aside. As regressors, in this method we train a set of linear models (LM) using the samples in the training set that are not identified as outliers and a given degree polynomial. The fitting process is done with a weighted least squares function, in which the weights are obtained by applying a likelihood function to the occurrence vector.

2.4. Probabilistic macro sampling

As opposed to the former methods detailed in Sections 2.2 and 2.3, no samples of the training set are discarded but rather they are assigned a likelihood of participation for training the model as a function of the occurrence vector v. We perform a random sampling with replacement with size equals to the size of elements in the training set, and the vector of probability weights for obtaining the elements of the sampled training set equals to the corresponding likelihood function. The new sampled set is used to build predictive models. We use neural networks as regressors considering the minimum mean squared error (MSE) in a number of replicates for each sampled set. This process is repeated due to the random nature of the method in order to assess the stability of the results.

2.5. Weighted regression linear model via PAELLA

In this method we fit a set of LM using all samples in the training set and a given degree polynomial. In this case, we use a traditional weighted regression and, as a novelty, we obtain the sample weights by applying a likelihood function to the occurrence vector of PAELLA algorithm without discarding outliers.

2.6. Weighted regression multilayer perceptron via PAELLA

In this method we train MLP models with the complete set of training samples using weighted regression. MSE error is optimized using as sample weights the values obtained by applying a likelihood function to the occurrence vector. The MLP models are trained using a SGD with momentum

method as training strategy and an early stopping method. An exhaustive grid search is performed in order to choose the optimal hyperparameters.

3. Experimental survey

3.1. Weighted regression via PAELLA vs. outlier removal and macro sampling

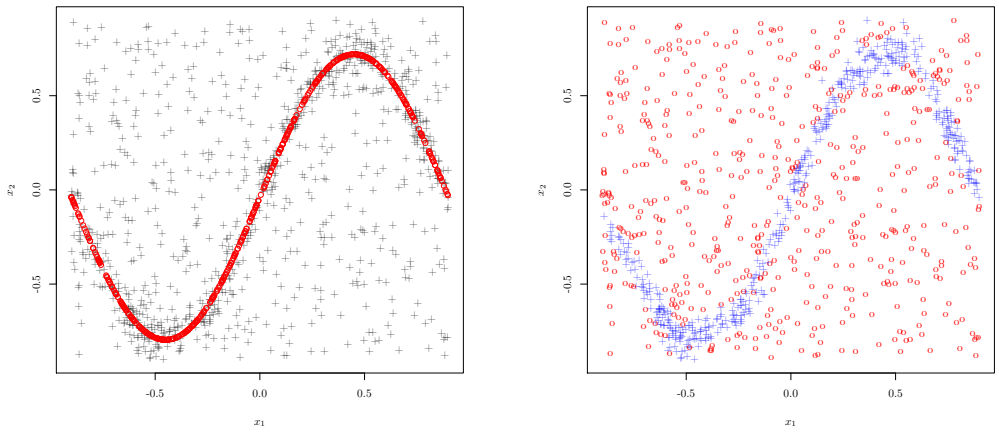
In this section, we compare the traditional use of PAELLA for outlier removal, and two novel approaches in which we use the information provided by PAELLA through the occurrence vector in order to perform probabilistic macro sampling regression, as well as weighted regression. We initially experimented with the dataset in [13] for training purposes and created a new test set for evaluation of the methods.

The artificial training set X of 1000 samples follows a sinusoidal function altered by some normal noise as shown in Fig. 1a. We created 500 samples using the formula in Eq. 1, and the rest using a uniform random distribution. We also created a test set of 500 samples that follows the equation in Eq. 1 as it is shown using red circles in Fig. 1a.

$$x_2 = \sin(x_1) + \epsilon \tag{1}$$

where $\epsilon \sim \mathcal{N}(\mu = 0,\ \sigma = 0.1)$ and each sample is featured with two variables (x_1, x_2). The dataset is highly noisy, hence a challenge for predicting a model.

We applied the PAELLA algorithm to the training set in order to obtain the values in the occurrence vector v, see Phase 3. Then, we computed some models on the training set for each of the three experiments performed.



(a) (b)

Figure 1: (a) Artificial dataset. The samples of the training set are represented with '+' symbols. The test set appears overlaid with red circles. (b) Outlier identification of the samples in the training set using PAELLA for a threshold of 99%. The red circles mark the samples identified as outliers while the blue '+' symbols show the samples that are not considered as outliers.

Finally, we evaluated the results on the test set by means of the MSE defined in Eq. 2.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

where y_i is the actual value of a sample in the dataset of n samples that follows the formula without noise and $\hat{y}_i$ is the corresponding predicted value. In this case, the actual values y_i are given by the pure signal of the artificial test set and the predicted values $\hat{y}_i$ by the output of the proposed models.

3.1.1. First experiment: outlier removal

In the first experiment, we obtained the list of potential outliers by comparing the occurrence vector v to a frequency threshold $t = 0.99$. Fig. 1b shows the result of applying PAELLA to the training set for a threshold of $t = 0.99$. It can be seen that the samples that were not identified as outliers mainly belong to the fuzzy sinusoidal function.

We considered both methods for outlier removal, using as regressors neural networks (Section 2.2) and linear models (Section 2.3).

For the neural networks, we performed the exhaustive grid search using all possible combinations of the following parameters: $[2, 3, 4, 5, 6]$ for the number of hidden neurons, $[0.001, 0.005, 0.010]$ for the learning rates, and $[0.001, 0.005, 0.010]$ for momentum. Thirty replicates were taken in order to evaluate the performance for different values of the initial weights and biases. This resulted in a total of 1350 neural networks. The best MLP model yielded a MSE equals to 0.0192.

For the linear models, a fifth degree polynomial was used. As a simple choice for the likelihood function, we considered a power function v^p where v is the occurrence vector and p a given power. This choice permits different degrees of penalization to the outlying samples in a simple manner by just varying the variable p.

We considered as powers, p, the set of $\{p|p \leqslant 100, p \in \mathbb{N}\}$, where $\mathbb{N}$ is the set of natural numbers. Results on the test set are presented in Table 1 under outlier removal column and Fig. 2 marked with red crosses. The best result was achieved for a power $p = 34$ with a MSE equals to 0.0157.

Table 1: Results from weighted regression vs. macro sampling vs. outlier removal using linear models.

Likelihood function	MSE		
	weighted regression	macro sampling	outlier removal
v^1	0.1051	0.0798	0.0203
v^2	0.0645	0.0653	0.0202
v^3	0.0458	0.0497	0.0182
v^4	0.0349	0.0267	0.0187
v^5	0.0277	0.0312	0.0185
v^{10}	0.0131	0.0237	0.0168

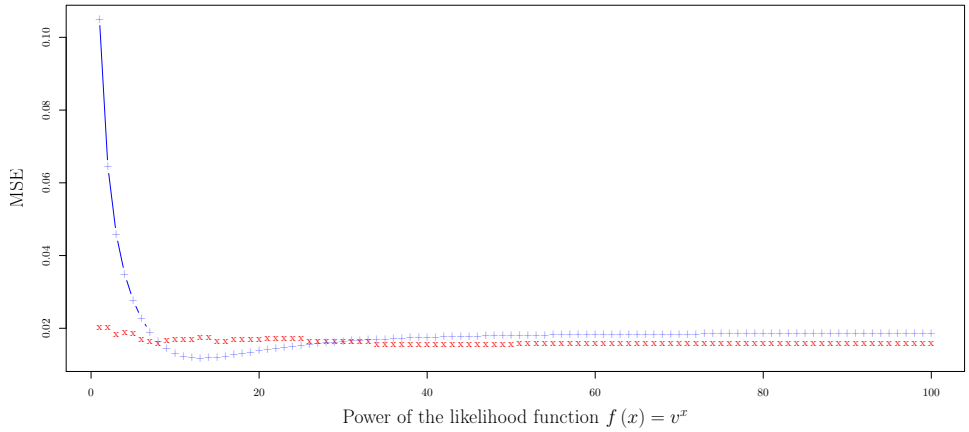


Figure 2: Mean squared error (MSE) obtained using linear models for the likelihood functions with powers in the set of natural numbers less or equal than 100. The red crosses represent the MSE for outlier removal method and the blue '+' signs indicate the MSE for weighted regression method.

3.1.2. Second experiment: probabilistic macro sampling

As for this second experiment, we followed the procedure of the method described in Section 2.4 and considered the following powers p of the occurrence vector as likelihood functions, v^p: $p = [1, 2, 3, 4, 5, 10]$.

For the neural networks, we chose 3 hidden neurons, 0.001 for the learning rates, and 0.001 for momentum, which proved to work successfully in the first experiment. Again, thirty replicates were taken in order to evaluate the performance for different values of the initial weights and biases. We performed 1000 repetitions considering the models that produced minimum MSEs and obtained the results on the test set shown in Table 1 under macro sampling column. The best result, MSE equals to 0.0237, was achieved for $p = 10$.

3.1.3. Third experiment: weighted regression via PAELLA

Regarding the third experiment, we followed the method in Section 2.5 using a fifth degree polynomial to fit the LM. We considered the powers, p, in the set $\{p|p \leqslant 100, p \in \mathbb{N}\}$. Table 1 partially shows the results on the test set under weighted regression column. The predicted outputs with this method are represented with blue circles in Fig. 3. Fig. 4 shows the MSE of the weighted regression method for the first 40 powers with blue '+' signs in comparison with outlier removal using MLP, and Fig. 2 illustrates the MSE of the weighted regression method in comparison with outlier removal using LM. The best result was achieved for a power $p = 13$ with a MSE equals to 0.0119.

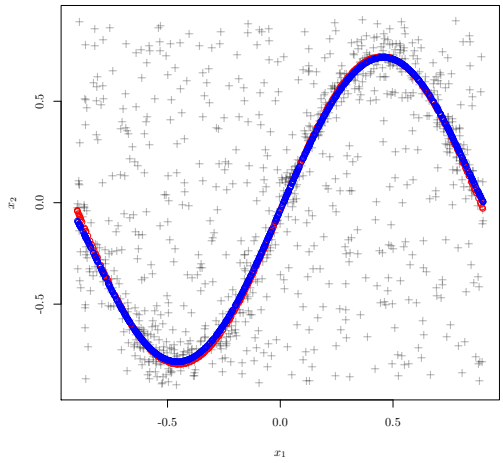


Figure 3: Prediction applying the weighted regression method. The samples of the training set are represented with '+' symbols. The test set appears overlaid with red circles. The predicted outputs with weighted regression method are overlaid with blue circles.

3.1.4. Comparison of the three experiments

As it can be seen in Table 1, Fig. 2 and Fig. 4, for small values of the power of the likelihood function up to $p = 9$, the outlier removal method outperformed the rest. However, for greater values of the power and up to $p = 28$, the weighted regression method yields the best results in comparison with the rest. The best overall performance was achieved using weighted regression with $p = 13$, obtaining a MSE equals to 0.0119. Therefore, weighted regression based on PAELLA, which allows the participation of all samples for building the model, is able to outperform the models fitted using only clean data and the macro sampling strategy presented in [13].

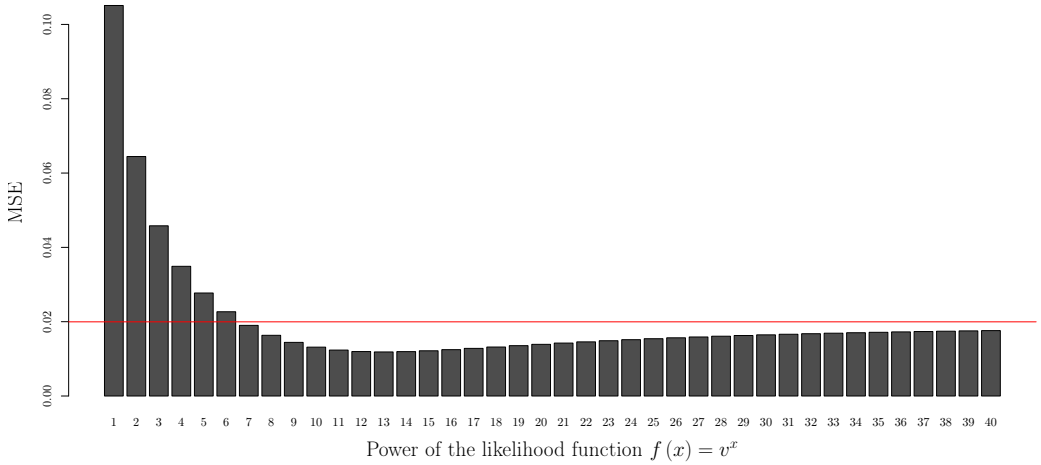


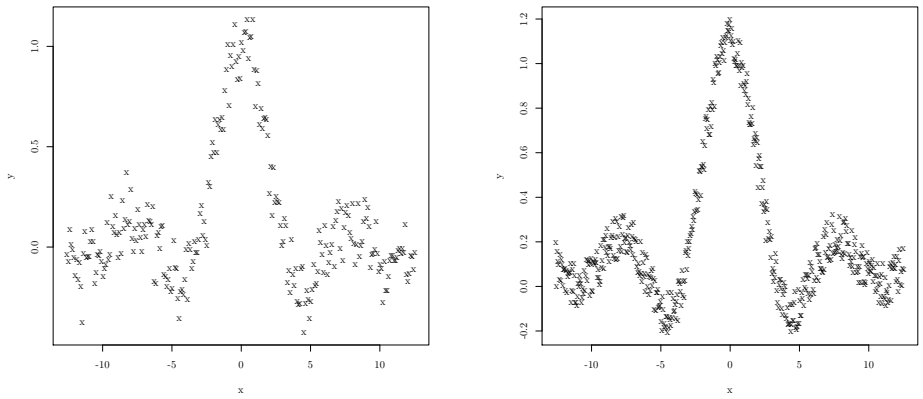
Figure 4: The bars of the barplot represent the MSE using the weighted regression LM method and likelihood functions with powers in the set of natural numbers less or equal than 100. The horizontal red line at height 0.0192 shows the MSE obtained by using the outlier removal method with MLP and threshold 0.99.

3.2. Weighted regression via PAELLA vs. state-of-the-art ϵ-TSVR

In this section, we compare the weighted regression via PAELLA using LM and MLP models, as described in Sections 2.5 and 2.6 respectively, to a state-of-the-art method presented by Shao et al. [16]. We describe the dataset used in the original work [16] and that we used as well for the comparison. We also provide details about the error metrics and results.

3.2.1. Dataset

Shao et al. [16] presented this dataset and an efficient support vector machine regression technique, named ϵ-TSVR. We used the publicly available



(a) (b)

Figure 5: Artificial dataset presented in [16] to evaluate ϵ-TSVR state-of-the-art regression technique. (a) Training set. (b) Test set.

dataset[2] published in [16]. The samples (x_i, y_i) of the dataset follow a cardinal sine (sinc) function polluted by Gaussian noise with 0 and 0.2 mean and standard deviation, respectively, see Eq. 3. The dataset is made of 252 training and 503 test samples that are shown in Fig. 5.

$$y_i = \frac{\sin(x_i)}{x_i} + \xi_i, \quad x \sim U[-4\pi, 4\pi], \quad \xi_i \sim N(0, 0.2^2) \tag{3}$$

3.2.2. Error metrics

We evaluated the method by means of the normalized mean square error (NMSE), mean absolute percentage error (MAPE), symmetric mean absolute percentage error (sMAPE) –Eq. 8–, and mean absolute scaled error (MASE) –Eq. 9–. We chose these metrics due to their popular use and because they

[2]Code available in http://www.optimal-group.org/Resource/WLETSVR.html

were employed to evaluate the original methods of reference ϵ-TSVR [16] and WL-ϵ-TSVR [17].

$$NMSE = \frac{1}{n}\sum_{i=1}^{n}\frac{(y_i - \hat{y}_i)^2}{\overline{y}\overline{\hat{y}}} \tag{4}$$

where

$$\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{5}$$

$$\overline{\hat{y}} = \frac{1}{n}\sum_{i=1}^{n} \hat{y}_i \tag{6}$$

y_i is the actual value of a sample in the dataset of n samples that follows the formula without noise and $\hat{y}_i$ is the corresponding predicted value. The same notation is kept for the rest of formulae in the paper.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i} \cdot 100 \tag{7}$$

$$sMAPE = \frac{2}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i + \hat{y}_i} \tag{8}$$

$$MASE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{|y_i - \hat{y}_i|}{\frac{1}{n-1}\sum_{i=2}^{n}|y_i - y_{i-1}|}\right) \tag{9}$$

3.2.3. Experimental setup

On the one hand, the set of linear models were configured with a fifth degree polynomial. On the other hand, we configured the MLP with one hidden layer using a hyperbolic tangent as activation function of the hidden neurons and the linear function as activation for the output layer. The exhaustive grid search was performed using the hyperparameters described in Table 2.

Table 2: Hyperparameters considered for exhaustive grid search for MLP weighted regression.

Parameter	Values
learning rate	$[1, 2, \ldots, 10]$
momentum	$[0.9, 0.09, 0.009]$
Nesterov	[True, False]
epochs	$[500, 1000, 5000]$
p	$[1, 2, \ldots, 10]$

Table 3: Results of the proposed weighted regression LM and MLP models vs. ϵ-TSVR state-of-the-art technique. In bold, the best results per error metric are highlighted.

Experiment	NMSE	MAPE	sMAPE	MASE
Weighted regression LM	**0.002967**	2.197	0.908	0.094
Weighted regression MLP	0.003049	**0.883**	**0.643**	**0.058**
ϵ-TSVR [16]	0.003044	2.557	0.931	0.100

3.2.4. Results

Results obtained with our proposed method using weighted regression with LM and MLP models and with ϵ-TSVR are presented in Table 3. Our method achieved better results for all considered metrics. MLP models outperformed LM for all but NMSE metric. We illustrate the predicted values using weighted regression with MLP models on top of the test samples in Fig. 6.

3.3. Weighted regression via PAELLA vs. state-of-the-art WL-ϵ-TSVR

In this section, we compare the weighted regression via PAELLA using LM and MLP models to a state-of-the-art method presented by Ye et al. [17].

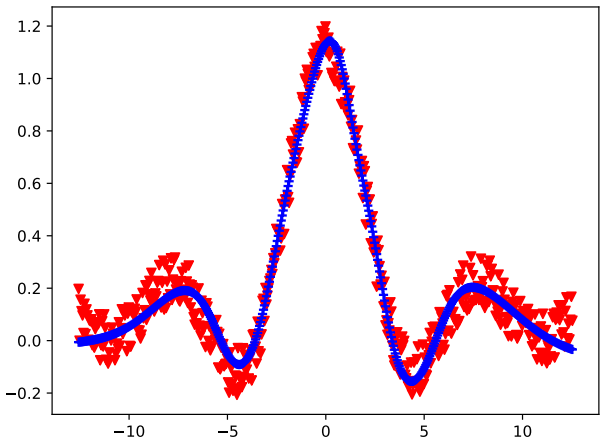


Figure 6: Prediction of weighted regression MLP using PAELLA for obtaining the sample weights on the dataset in [16]. The samples of the test set are shown with a red triangle marker. The predicted values are shown as blue '+' symbols.

We used the same error metrics and experimental setup as in the comparison with Shao et al. [16] for ϵ-TSVR in Section 3.2. Below, we describe the dataset used in the original work [17] and that we used as well for the comparison and we provide the results obtained.

3.3.1. Dataset

Ye et al. [17] proposed an improved version of the ϵ-TSVR method, named WL-ϵ-TSVR, that is based on weighted Lagrange support vector regression. The dataset at hand was also introduced in this work in order to evaluate the method.

We generated 1000 training samples following Eq. 10 and polluted by Gaussian noise with mean 0 and standard deviation 0.1, and one outlier with deviation of -1, as in the referred paper [17]. Similarly, 1000 test samples

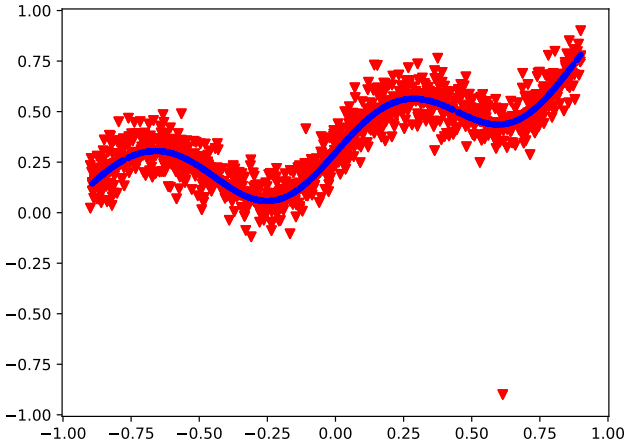


Figure 7: Prediction of weighted regression MLP models using PAELLA for obtaining the sample weights on the dataset in [17]. The samples of the test set are shown with a red triangle marker. The predicted values are shown as blue circles which look like a continuous curve.

were created following the same procedure, they are shown with red triangle markers in Fig. 7.

$$y = 0.2 \sin(2\pi x) + 0.2x^2 + 0.3, \quad x \in [0, 2] \tag{10}$$

3.3.2. Results

The results using weighted regression via PAELLA with LM and MLP models and with WL-ϵ-TSVR are presented in Table 4. Our method achieved better results for all considered metrics. MLP models outperformed LM for all but MASE metric. The proposed methods yielded better results than WL-ϵ-TSVR for every considered metric. The predicted values using weighted regression with MLP models are overlaid with blue circles on top of the test

Table 4: Results of the proposed weighted regression (WR) model using linear models (LM) and multilayer perceptron models (MLP) vs. WL-ϵ-TSVR state-of-the-art technique. The time is shown as mean and standard deviation in a sufficient number of loops. In bold, the best results per error metric are highlighted.

Experiment	NMSE	MAPE	sMAPE	MASE	Time (ms)
WR LM	0.000116	0.017	0.016	**0.009**	**0.0193 ± 0.0000882**
WR MLP	**0.000030**	**0.010**	**0.010**	0.010	5.9 ± 0.0285
WL-ϵ-TSVR [17]	0.2211	-	0.1855	2.1766	448.1 ± 19.2

samples in Fig. 7.

In order to provide an analysis of the execution time, we computed the times required for predicting the output of the test set in terms of mean and standard deviation on a sufficient number of loops and repetitions. No less than 70 runs were considered for the averaged results. The available code for WL-ϵ-TSVR is provided in MATLAB and thus the time was computed using MatLab R2016a, whereas the time for both weighted regressions methods was computed using Python 3.6. It can be observed that the proposed methods, weighted regression MLP and weighted regression LM, are around 75 and 23000 times faster with respect to WL-ϵ-TSVR, respectively.

Considering performance both in terms of accuracy and execution times, weighted regression MLP should be used in occasions in which the accuracy is essential, whereas weighted regression LM should be used when a low execution time is required at the cost of losing a bit of accuracy.

4. Conclusions

The PAELLA algorithm for outlier identification and data cleaning has proven to be versatile enough to be useful as well in the context of robust regression. The results reported in this paper confirm that the occurrence vector provided by the PAELLA algorithm can boost the fitting of predictive models with an improvement rate in some experiments measured in tens of thousands. Moreover, this improvement is achieved without the need of discarding those samples otherwise marked as outliers. Among the different strategies reported, using the occurrence vector values —or a transformation of those through a custom function— as sample weights in weighted regression showed promising results both against common practice techniques such as outlier removal and state-of-the-art algorithms. Specifically, the proposed methods achieved a reduction of at least 2.53% and 94.61% for all error metrics in regard to ϵ-TSVR and WL-ϵ-TSVR, respectively. Moreover, weighted regression linear models and weighted regression multilayer perceptron models are about 23000 and 75 times faster than WL-ϵ-TSVR, respectively. The use of weighted regression multilayer perceptron is recommended when accuracy is critical whereas weighted regression linear models can be used to get a good compromise between accuracy and execution time.

Acknowledgements

We gratefully acknowledge the financial support of Spanish *Ministerio de Economía, Industria y Competitividad* through grant DPI2016-79960-C3-2-P. We would like to also express our gratitude to SCAYLE Castilla y León Supercomputing Center.

References

[1] C. Menéndez, J. Ordieres, F. Ortega, Importance of information preprocessing in the improvement of neural network results, Expert Systems 13 (2) (1996) 95–103.

[2] J. Ordieres, E. Vergara, R. Capuz, R. Salazar, Neural network prediction model for fine particulate matter (pm 2.5) on the us–mexico border in el paso (texas) and ciudad juárez (chihuahua), Environmental Modelling & Software 20 (5) (2005) 547–559.

[3] E. Salazar-Ruiz, J. Ordieres, E. Vergara, S. F. Capuz-Rizo, Development and comparative analysis of tropospheric ozone prediction models using linear and artificial intelligence-based models in mexicali, baja california (mexico) and calexico, california (us), Environmental Modelling & Software 23 (8) (2008) 1056–1069.

[4] B. Gong, J. Ordieres-Meré, Prediction of daily maximum ozone threshold exceedances by preprocessing and ensemble artificial intelligence techniques: Case study of hong kong, Environmental Modelling & Software 84 (2016) 290–303.

[5] G. Bing, J. Ordieres-Meré, C. B. Cabrera, Prediction models for ozone in metropolitan area of mexico city based on artificial intelligence techniques, International Journal of Information and Decision Sciences 7 (2) (2015) 115–139.

[6] Z. Lv, J. Chirivella, P. Gagliardo, Bigdata oriented multimedia mobile

health applications, Journal of Medical Systems 40 (5) (2016) 120. doi:10.1007/s10916-016-0475-8.

[7] J. Ordieres-Meré, F. Martínez-de Pisón-Ascacibar, A. González-Marcos, I. Ortiz-Marcos, Comparison of models created for the prediction of the mechanical properties of galvanized steel coils, Journal of Intelligent manufacturing 21 (4) (2010) 403–421.

[8] A. Gonzalez-Marcos, F. Alba-Elias, M. Castejon-Limas, J. Ordieres-Mere, Development of neural network-based models to predict mechanical properties of hot dip galvanised steel coils, International Journal of Data Mining, Modelling and Management 3 (4) (2011) 389–405.

[9] J. Ordieres, L. López, A. Bello, A. Garcia, Intelligent methods helping the design of a manufacturing system for die extrusion rubbers, International Journal of Computer Integrated Manufacturing 16 (3) (2003) 173–180.

[10] T. Dasu, T. Johnson, Exploratory Data Mining and Data Cleaning, Wiley, 2003. doi:10.1002/0471448354.fmatter.

[11] B. Walczak, Neural networks with robust backpropagation learning algorithm, Analytica Chimica Acta 322 (1) (1996) 21–29. doi:10.1016/0003-2670(95)00552-8.

[12] M. C. Limas, J. B. O. Meré, F. J. M. D. P. Ascacibar, E. P. V. González, Outlier detection and data cleaning in multivariate non-normal samples: The PAELLA algorithm, Data Mining and Knowledge Discoverydoi:10.1023/B:DAMI.0000031630.50685.7c.

[13] M. Castejón-Limas, H. Alaiz-Moreton, L. Fernández-Robles, J. Alfonso-Cendón, C. Fernández-Llamas, L. Sánchez-González, H. Pérez, Coupling the paella algorithm to predictive models, in: H. Pérez García, J. Alfonso-Cendón, L. Sánchez González, H. Quintián, E. Corchado (Eds.), International Joint Conference SOCO'17-CISIS'17-ICEUTE'17 León, Spain, September 6–8, 2017, Proceeding, Springer International Publishing, Cham, 2018, pp. 505–512.

[14] M. Castejón-Limas, H. Alaiz-Moreton, L. Fernández-Robles, J. Alfonso-Cendón, C. Fernández-Llamas, L. Sánchez-González, H. Pérez, Paella as a booster in weighted regression, in: H. Pérez García, J. Alfonso-Cendón, L. Sánchez González, H. Quintián, E. Corchado (Eds.), International Joint Conference SOCO'17-CISIS'17-ICEUTE'17 León, Spain, September 6–8, 2017, Proceeding, Springer International Publishing, Cham, 2018, pp. 259–265.

[15] N. Qian, On the momentum term in gradient descent learning algorithms, Neural Networks 12 (1) (1999) 145 – 151. doi:https://doi.org/10.1016/S0893-6080(98)00116-6.

[16] Y.-H. Shao, C.-H. Zhang, Z.-M. Yang, L. Jing, N.-Y. Deng, An ϵ-twin support vector machine for regression, Neural Computing and Applications 23 (1) (2013) 175–185. doi:10.1007/s00521-012-0924-3.

[17] Y.-F. Ye, L. Bai, X.-Y. Hua, Y.-H. Shao, Z. Wang, N.-Y. Deng, Weighted lagrange ϵ-twin support vector regression, Neurocomputing 197 (2016) 53 – 68. doi:https://doi.org/10.1016/j.neucom.2016.01.038.

Manuel Castejón-Limas is an associate professor at Universidad de León. He is a Master in Engineering by Universidad de Oviedo (1999) and PhD by Universidad de La Rioja (2004). His research interests are related with data science applications in project management, factory process optimization and environmental modeling.

Hector Alaiz-Moreton received his degree in Computer Science, performing the final project at Dublin Institute of Technology, in 2003. He received his PhD in Information Technologies in 2008 (University of Leon). He has worked like a lecturer since 2005 at the School of Engineering at the University of Leon. His research interests include knowledge engineering, machine and deep learning, networks communication and security. He has several works published in international conferences, as well as books and scientific papers in peer review journals. He has been member of scientific committees in conferences. He has headed several PhD Thesis and research projects.

Laura Fernández-Robles is an assistant lecturer and researcher at University of León. In 2016, she received the Ph.D. degree from University of Groningen, the Netherlands and from University of León, Spain. In 2011, she received the M.Sc. degree in Intelligent Systems in Engineering and in 2009 in Industrial Engineering both at University of León. She has participated in 1 European project and 4 Spanish projects. Her current research interests

include computer vision, pattern recognition and data science applied to industrial, cyber-security and medical problems.

Javier Alfonso-Cendón is an Associate Professor in the Area of Engineering Projects at the Universidad de Leon. He is PhD in Engineering and Computer Engineer by the University of León. He has many specialized courses in the field of engineering and communication technologies. He has more than 20 national and international publications in the field of ICT.

Camino Fernández-Llamas received her degree in Computer Science, an M.S. degree in Artificial Intelligence and a Ph.D. in Computer Science from Universidad Politécnica de Madrid. She started teaching at Universidad Carlos III de Madrid and moved later to Universidad de León. She spent the school year 2014-2015 at the School of Computer Science of the Carnegie Mellon University (USA). Since 2015 she has been affiliated to the Research Institute on Applied Sciences of Cybersecurity where she is in charge of secure software development. Her main interests include programming, haptic simulation, e-learning and secure coding.

Lidia Sánchez-González got the Computing Engineering degree from the University of León (Spain) in 2002, and the Ph.D. from the University of León in 2007. Since 2003, she has been working at the University of León (Spain) as a Lecturer in the area of Computer Architecture and Technology in the Department of Mechanical, Computing and Aerospace Engineering. Her research interests focus on digital image processing and analysis applied to medical images and industrial processes. She is also interested in high performance computing in order to increase the performance of a system.

Hilde Pérez is Associate Professor and head of the Department of Mechanical, Computer and Aerospace Engineering at the University of Leon. She received her engineering degree in Mechanical Engineering from the University of Oviedo and in Electrical and Electronic Engineering from the University of León. She received her Ph. D. from Polytechnic University of Madrid, obtaining the Outstanding Doctorate Award in 2012. She has been involved in different national research projects in collaboration with the Polytechnic University of Madrid. The research areas of interest are related with smart systems for manufacturing and collaborative robots for manufacturing industry, modelling and simulation of machining processes, micromanufacturing and high performance machining.