Journal of Computational and Applied Mathematics 443 (2024) 115773

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

An entropy-based approach for a robust least squares spline approximation

Luigi Brugnano [a], Domenico Giordano [b], Felice Iavernaro [c,*], Giorgia Rubino [c]

[a] Dipartimento di Matematica e Informatica "U. Dini", Università di Firenze, Italy
[b] ESTEC (retired), European Space Agency, Noordwijk, The Netherlands
[c] Dipartimento di Matematica, Università degli Studi di Bari Aldo Moro, Italy

ARTICLE INFO

MSC:
65D10
94A17

Keywords:
Weighted least squares approximation
Robust regression
B-splines
Entropy
Outliers detection
Data smoothing

ABSTRACT

We consider the weighted least squares spline approximation of a noisy dataset. By interpreting the weights as a probability distribution, we maximize the associated entropy subject to the constraint that the mean squared error is prescribed to a desired (small) value. Acting on this error yields a robust regression method that automatically detects and removes outliers from the data during the fitting procedure, by assigning them a very small weight. We discuss the use of both spline functions and spline curves. A number of numerical illustrations have been included to disclose the potentialities of the maximal-entropy approach in different application fields.

1. Introduction

With the advent of computer-aided modern technology, sheer volumes of data need to be preprocessed in order to make them suitable for the subsequent data-driven tasks they are intended for. Real data are often affected by various imperfections, including noise, poor sampling, missing values and outliers. The automatic identification and removal of these inconsistencies has become of paramount importance during the preprocessing phase of data, since they may significantly affect the predictive accuracy and efficiency of models such as those based upon single and multivariate regression, as well as of pattern recognition procedures resulting from machine learning and deep learning processes [1–4].

Identification of corrupted data also play a fundamental role in automatic anomaly detection, meant as the appearance of events or observations which are inconsistent with the pattern underlying a given dataset. Anomaly detection has become increasingly important in many application areas ranging from statistics, cyber security, medicine, event detection in sensor networks, financial fraud and machine learning [5].

Outliers may be thought of extreme values that deviate significantly from the trend defined by the majority of the data points, possibly due to errors or rare events, and that can consequently worsen the performance of many data analysis algorithms. Classical outlier detection methods often rely on specific assumptions about the data's distribution. However, in many real-world scenarios, estimating such a distribution beforehand can be challenging due to the data's dependence on various unknown or complex factors and the presence of highly noisy sources. This limitation becomes apparent in vast collections of time series data, especially within environmental investigation. Such a topic has recently garnered extensive research attention, especially in understanding the correlation between climate changes and the increasing severity of natural disasters [6,7].

* Corresponding author.
E-mail addresses: luigi.brugnano@unifi.it (L. Brugnano), dg.esa.retired@gmail.com (D. Giordano), felice.iavernaro@uniba.it (F. Iavernaro), g.rubino33@studenti.uniba.it (G. Rubino).

https://doi.org/10.1016/j.cam.2024.115773
Received 28 September 2023; Received in revised form 7 December 2023
Available online 12 January 2024

Extending the study addressed in [8] for the polynomial case, the present paper introduces a robust regression technique for spline approximation of both univariate and multivariate time series, considering scenarios where observations exhibit varying degrees of reliability (see [9,10] for recent related studies).

The use of splines in place of polynomials in data fitting problems is advantageous in many respects. As piecewise-defined functions, splines provide localized fits to different parts of the data. This aspect is crucial when the data exhibit complex patterns or variations. Unlike a single polynomial, which may struggle to capture abrupt changes in the data, splines inherently adapt to such variations without requiring an excessively high degree. Conversely, the use of a high-degree polynomial can lead to ill-conditioning issues and results in elevated computational costs.

The application of splines for managing data affected by noise has been widely explored across various statistical and related fields. Among the most prominent instances, we mention (cubic) smoothing splines and their multivariate version, thin-plate splines, that minimize a convex combination of the residual sum of squares and a smoothness penalty based on the second derivatives of the function [11,12].[1] A relevant advantage of these techniques is that they only require the selection of a single parameter, namely the smoothing coefficient realizing the convex combination, and one has not to care about the number and choice of knots; in fact these are placed at the unique predictors t_i.

In statistics, robust regression tries to overcome the limitations of the ordinary least squares when its underlying assumptions are violated, for example, due to the presence of outliers [13–16]. Robust algorithms widely used to handle data affected by both Gaussian noise and outliers include RLOWESS (Robust Locally Weighted Scatterplot Smoothing) and RLOESS (Robust Locally Estimated Scatterplot Smoothing) [17–19]. Both algorithms employ locally-weighted polynomial regression. A completely different approach is implemented in the RANSAC (RANdom SAmple Consensus) algorithm, a quite robust and flexible iterative method that estimates the parameters of a mathematical model, in our context a spline, from a set of observed data that contains outliers [20].

Smoothing splines, RLOWESS, RLOWESS and RANSAC algorithms are further described in Section 4, where they have been used for comparison purposes (see Example 4).

The procedure proposed in this paper tackles the challenges posed by outliers and noise by formulating a weighted least squares problem that leverages the statistical concept of entropy. To this end, we adopt the normalization condition that the weights sum to one, which allows us to interpret them as a probability distribution.

In more detail, to mitigate the negative influence of outliers and noise on the resulting approximating curve, the procedure maximizes the entropy H associated with the weight distribution, under the constraint that the resulting weighted mean squared error takes a prescribed value lower than the one corresponding to a uniform-weight distribution. Such a value may be either provided by the user, on the basis of what he would expect in absence of corrupted data, or automatically detected during the implementation of the procedure.

To better elucidate the role played here by entropy, we quote Jaynes [21, page 97]:

> ... the distribution that maximizes H, subject to constraints which represent whatever information we have, provides the most honest description of what we know. The probability is, by this process, spread out as widely as possible without contradicting the available information.

Translating Jaynes' words in our context, we may stress that the proposed approach ensures that as many data points as possible carry non-negligible weights, which results in maximizing the inlier set while adhering to the mean squared error constraint. To achieve this, the strategy assigns smaller weights to points that are more likely to be considered outliers, effectively minimizing their influence on defining the final shape of the approximating spline curve. It is important to note that this weighting task is seamlessly integrated into the fitting procedure, resulting in a unified methodology that eliminates the need for a preprocessing phase. Similarly to the RANSAC algorithm [20], the entropy-based approach proves particularly effective in handling situations where a substantial portion of the data is corrupted. However, unlike the RANSAC algorithm, it boasts the advantage of being deterministic in nature. Furthermore, by reinterpreting the weights as probabilities, we can readily justify the use of entropy as a mathematical tool for effectively handling corrupted data points.

The paper is structured as follows: In Section 2, we review the fundamental concepts related to weighted least squares spline approximation and introduce the corresponding notations. Section 3 presents a formal definition of the approximation problem using the entropy-based tool and proposes a simple algorithm to obtain the optimal solution for the constrained optimization problem. To demonstrate the functionality of the entropy tool, a few numerical illustrations are provided in Section 4. In Section 5, three examples involving real-world data are considered. Finally, in Section 6, we draw conclusions based on the findings.

2. Background

Consider a parametrized sequence of points $\{(t_i, y_i)\}_{i=1}^m$, where $t = (t_1, \dots, t_m)^\top$ is a non-decreasing sequence of real parameters and $y_i \in \mathbb{R}^s$ the corresponding data points. In the statistics parlance the sequence $\{(t_i, y_i)\}$ is often referred to as a multivariate time series. As is usual in this context, we introduce a change of variable that normalizes the data in $[0,1] \times [0,1]^s$[2]:

$$t_i \to \frac{t_i - t_{\min}}{t_{\max} - t_{\min}}, \qquad y_i \to \frac{y_i - y_{\min}}{y_{\max} - y_{\min}}.$$

[1] A further generalization and widely used smoothing techniques are penalized splines.

[2] In the sequel, all the operations and functions evaluations involving vectors are meant componentwise. For example, for a given vector $z = (z_1, \dots, z_k)^\top$ and a function $g : \mathbb{R} \to \mathbb{R}$, we have $g(z) = (g(z_1), \dots, g(z_k))^\top$.

where

$$t_{\min} = \min_{1\le i\le m} t_i, \quad t_{\max} = \max_{1\le i\le m} t_i$$

and, denoting by $y_i(j)$ the jth entry of the vector y_i,

$$y_{\min}(j) = \min_{1\le i\le m} y_i(j), \quad y_{\max}(j) = \max_{1\le i\le m} y_i(j), \quad j = 1,\dots,s.$$

Of course, one can revert to the original coordinates by employing the inverse transformations. We wish to fit the given data set by means of a spline curve f of degree d expanded along a B-spline basis $\{B_{j,d}(x)\}_{j=1}^{n}$, namely

$$f(x,c) = \sum_{j=1}^{n} c_j B_{j,d}(x). \tag{1}$$

Here, $c = (c_1^\top,\dots,c_n^\top)^\top \in \mathbb{R}^{sn}$ is a set of n control points, each of length s, and the B-splines $B_j(x)$ are defined on a non-decreasing sequence of $(d+1)$-regular knots

$$0 = x_1 = \cdots = x_{d+1} < x_{d+2} \le \dots \le x_n < x_{n+1} = \cdots = x_{n+d+1} = 1, \tag{2}$$

via the three-terms recursive relation[3]

$$B_{j,d}(x) = \frac{x - x_j}{x_{j+d} - x_j} B_{j,d-1}(x) + \frac{x_{j+d+1} - x}{x_{j+d+1} - x_{j+1}} B_{j+1,d-1}(x),$$

with

$$B_{j,0}(x) = \begin{cases} 1, & \text{if } x_j \le x < x_{j+1}, \\ 0, & \text{otherwise.} \end{cases}$$

Besides the conditions at the end points in (2), the $(d+1)$-regularity of the knot vector also imposes that $n \ge d+1$ and $x_j < x_{j+d+1}$, for $j = 1,\dots,n$, which are relevant assumptions for the B-splines linear independence property [22]. In the sequel, for sake of simplicity, we will omit the second subscript in $B_{j,d}(x)$.

Now, for a given vector $w = (w_1,\dots,w_m)^\top$ of (positive) weights satisfying the normalization condition

$$\sum_{i=1}^{m} w_i = 1, \tag{3}$$

we consider the weighted mean squared error

$$\overline{E^2} = \sum_{i=1}^{m} w_i \|f(t_i,c) - y_i\|_2^2 \tag{4}$$

as an estimate of the approximation accuracy of a spline function $f(x,c)$ in the form (1) to the given data set. Denoting by I_s the identity matrix of dimension s and introducing the generalized Vandermonde matrix

$$A = \begin{pmatrix} B_1(t_1) & \cdots & B_n(t_1) \\ \vdots & & \vdots \\ B_1(t_m) & \cdots & B_n(t_m) \end{pmatrix} \in \mathbb{R}^{m\times n},$$

the vector $y = (y_1^\top,\dots,y_m^\top)^\top \in \mathbb{R}^{sm}$ and the diagonal matrix $W = \mathrm{diag}(w_1,\dots,w_m)$, (4) may be cast in two equivalent forms that will be conveniently exploited for calculation and implementation purposes:

$$\begin{aligned} \overline{E^2} &= (f(t,c) - y)^\top (W \otimes I_s)(f(t,c) - y) \\ &= \|(\sqrt{W} \otimes I_s)(f(t,c) - y)\|_2^2 \\ &= \|(\sqrt{W} \otimes I_s)((A \otimes I_s)c - y)\|_2^2 \end{aligned} \tag{5}$$

and, denoting by $e_s = (1,\dots,1)^\top$ the unit vector of length s,

$$\begin{aligned} \overline{E^2} &= (w \otimes e_s)^\top (f(t,c) - y)^2 \\ &= (w \otimes e_s)^\top ((A \otimes I_s)c - y)^2. \end{aligned} \tag{6}$$

For a prescribed choice of weights, the *weighted least squares* (WLS) *approximation problem* consists in finding the (vector) coefficients c_j such that the corresponding weighted mean squared error (5) is minimized. As is well known, differentiating (5) with respect to c, this requirement leads to the normal system

$$(A^\top W A \otimes I_s)c = (A^\top W \otimes I_s)y, \tag{7}$$

which results from computing the stationary points of $\overline{E^2}$ regarded as a function of c.

[3] If a division by zero occurs, the related term is neglected.

Under the assumption that for any $j = 1, \dots, n$ a t_{i_j} exists such that $B(t_{i_j}) \neq 0$, matrix $A^\top W A$ is positive definite and the Cholesky factorization may be employed to transform (7) into a couple of triangular systems. More in general, also to prevent a worsening of the conditioning, one avoids the left multiplication by the matrix $A^\top$ and directly deals with the least squares solution of the overdetermined system

$$(\sqrt{W}A \otimes I_s)c = (\sqrt{W} \otimes I_s)y. \tag{8}$$

In such a case, application of the QR factorization algorithm with column pivoting, or the SVD decomposition to the rectangular matrix $\sqrt{W}A$ may be considered to solve the associated least squares problem.

Remark 1. In the event that the components $y_i(j)$, $j = 1, \dots, s$ are affected by sources of noise of different size depending on j, one could improve (4) by allowing a different weight for each component of the error $f(t_i, c) - y_i$. This is tantamount to consider a vector of weights w of length ms and the related mean squared error defined as

$$\overline{E^2} = w^\top (f(t, c) - y)^2 \equiv \|\sqrt{W}(f(t, c) - y)\|_2^2, \tag{9}$$

with $W = \mathrm{diag}(w)$. In the numerical tests discussed in Sections 4 and 5 both approaches showed pretty similar results, so we only included those relying on (4).

In the sequel, $\overline{E^2}_{\mathrm{uw}}$ will denote the mean squared error resulting from the ordinary least squared (OLS) approximation defined on the uniform-weight distribution $w_i = 1/m$, namely

$$\overline{E^2}_{\mathrm{uw}} = \frac{1}{m}\sum_{i=1}^{m}(f(t_i, \bar{c}) - y_i)^2, \tag{10}$$

where $\bar{c}$ satisfies the normal linear system (7) with $W = I_m/m$, I_m being the identity matrix of dimension m.

3. Maximum entropy weighted least squares spline approximation

The use of a weighted mean squared error is helpful when the data highlight different level of accuracies, due to the presence of noise and/or outliers. In such a case, it would be appropriate to attach large weights to very accurate data points and small weights to data points which are most likely affected by a high level of inaccuracy. In fact, a weight w_i approaching zero makes the corresponding data point y_i irrelevant for the purpose of the fitting procedure. On the other hand, increasing the magnitude of w_i will make $f(t_i, c)$ closest to y_i. It turns out that, under the normalization condition (3), the WLS approximation will mimic the OLS one applied to the subset of data carrying relatively large weights.

By exploiting an entropy-based argument, the *maximum entropy weighted least squares* (MEWLS) approximation tries to devise an automatic, easy-to-understand and effective procedure for assigning the correct weight to each data point during the fitting procedure. The MEWLS approach based on splines approximating functions in the form (1) is defined by the following set of equations (e_m stands for the unit vectors of length m):

$$\begin{array}{ll} \text{maximize} & -w^\top \log w, \\ \text{subject to:} & w^\top e_m = 1, \\ & (w \otimes e_s)^\top (f(t, c) - y)^2 = \overline{E^2}. \end{array} \tag{11}$$

In other words, we wish to maximize the entropy function

$$H(w) = -w^\top \log w = -\sum_{i=1}^{m} w_i \log w_i \tag{12}$$

associated with a weight distribution w satisfying the normalization condition $\sum_i w_i = 1$, subject to the constraint that the corresponding mean squared error attains a prescribed value $\overline{E^2}$.

As is well known, problem (11), deprived of the second constraint, admits the solution $w_i = 1/m$, which leads us back to the ordinary least squares problem with uniform weights and associated means squared error $\overline{E^2}_{\mathrm{uw}}$. Clearly, the very same solution is obtained when solving the complete set of equations in (11) under the choice $\overline{E^2} = \overline{E^2}_{\mathrm{uw}}$, so (11) contains the ordinary least squares problem as a special instance. By setting $\overline{E^2}$ to a suitable value lower than the mean squared error $\overline{E^2}_{\mathrm{uw}}$, the weights selection technique based upon the maximal-entropy argument epitomized by (11) is aimed at mitigating the effect of outliers and noise in the data while solving the weighted least squares problem. To highlight the relation between $\overline{E^2}$ and $\overline{E^2}_{\mathrm{uw}}$, we assume in the sequel

$$\overline{E^2} = \frac{1}{r}\overline{E^2}_{\mathrm{uw}} \tag{13}$$

where $r > 1$ is a suitable reduction factor.

According to the Lagrange multiplier theorem, we compute the stationary points of the Lagrangian function

$$\mathcal{L}(w, c, \lambda_1, \lambda_2) = w^\top \log w + \lambda_1(w^\top e_m - 1) + \lambda_2\left((w \otimes e_s)^\top (f(t, c) - y)^2 - \overline{E^2}\right). \tag{14}$$

Differentiating, we get:

$$\frac{\partial \mathcal{L}}{\partial w} = e_m + \log w + \lambda_1 e_m + \lambda_2\left((I_m \otimes e_s)^\top (f(t, c) - y)^2\right),$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial c} &= 2\lambda_2\left((A^\top W A\otimes I_s)c-(A^\top W\otimes I_s)y\right),\\ \frac{\partial \mathcal{L}}{\partial \lambda_1} &= w^\top e_m-1,\\ \frac{\partial \mathcal{L}}{\partial \lambda_2} &= (w\otimes e_s)^\top(f(t,c)-y)^2-\overline{E^2}.\end{aligned} \tag{15}$$

The last term in (15) is the vector of length m

$$\lambda_2\left(\|f(t_1,c)-y_1\|_2^2,\ldots,\|f(t_m,c)-y_m\|_2^2\right)^\top, \tag{16}$$

while (15) comes from the equivalence of formulae (5) and (6), after observing that the first two terms in the Lagrangian (14) do not depend on the spline coefficients c_i. The stationary points of $\mathcal{L}$ are the solutions of the following set of $n+m+2$ equations in as many unknowns $c\in\mathbb{R}^n$, $w\in\mathbb{R}^m$, λ_1 and λ_2:

$$(A^\top W A\otimes I_s)c-(A^\top W\otimes I_s)y = 0, \tag{17}$$

$$(w\otimes e_s)^\top(f(t,c)-y)^2-\overline{E^2} = 0, \tag{18}$$

$$e_m+\log w+\lambda_1 e_m+\lambda_2\left((I_m\otimes e_s)^\top(f(t,c)-y)^2\right) = 0, \tag{19}$$

$$w^\top e_m-1 = 0. \tag{20}$$

By exploiting the weights normalization condition (20), we can easily remove the unknown λ_1. To this end, we first recast Eq. (19) as

$$w=\exp(-(1+\lambda_1))\cdot\exp\left(-\lambda_2\left((I_m\otimes e_s)^\top(f(t,c)-y)^2\right)\right).$$

Multiplying both sides by $e_m^\top$ and taking into account (16) and (20) yields

$$1=\exp(-(1+\lambda_1))\cdot Q(c,\lambda_2),\qquad \text{with } Q(c,\lambda_2)=\sum_{i=1}^m\exp\left(-\lambda_2\|f(t_i,c)-y_i\|_2^2\right)$$

and hence

$$w=\frac{1}{Q(c,\lambda_2)}\cdot\exp\left(-\lambda_2\left((I_m\otimes e_s)^\top(f(t,c)-y)^2\right)\right) \tag{21}$$

that will replace (19) and (20). Plugging (21) into (18) we arrive at the final shape of the system to be solved:

$$(A^\top W A\otimes I_s)c-(A^\top W\otimes I_s)y = 0, \tag{22}$$

$$\sum_{i=1}^m\|f(t_i,c)-y_i\|_2^2\cdot\exp\left(-\lambda_2\|f(t_i,c)-y_i\|_2^2\right)-\sum_{i=1}^m\exp\left(-\lambda_2\|f(t_i,c)-y_i\|_2^2\right)\overline{E^2} = 0, \tag{23}$$

$$w-\frac{1}{Q(c,\lambda_2)}\cdot\exp\left(-\lambda_2\left((I_m\otimes e_s)^\top(f(t,c)-y)^2\right)\right) = 0. \tag{24}$$

Before facing the question of how to solve the system numerically, a few remarks are in order:

- (22) is nothing but the normal linear system one would get when handling the least squares problem with constant weights (see (7)). It can be therefore expressed as the overdetermined system (8) which has to be solved in the least squares sense;
- (23) is a scalar equation that, for a given vector c, may be easily solved with respect to the Lagrange multiplier λ_2 via a Newton or Newton-like iteration;
- Eq. (24) is explicit with respect to the unknown w, for given λ_2 and c.

Therefore, a quite natural technique to solve the nonlinear system (22)–(24) is yielded by the hybrid iteration summarized in Algorithm 1 (*tol* is an input tolerance for the stopping criterion and $||\cdot||$ denotes any vector norm).

Algorithm 1: Numerical procedure for solving system (22)–(23).

initially, set $W^{(0)}\leftarrow\frac{1}{m}I_m$, $\lambda_2^{(0)}\leftarrow 0$; $k\leftarrow 0$;
repeat:
  $k\leftarrow k+1$;
  $c^{(k)}\leftarrow\underset{c}{\operatorname{argmin}}\,||(\sqrt{W^{(k-1)}}A\otimes I_s)c-(\sqrt{W^{(k-1)}}\otimes I_s)y||_2$;
  employ a Newton iteration scheme with initial guess $c^{(k)},\lambda_2^{(k-1)}$, to solve (23) and get $\lambda_2^{(k)}$;
  $w^{(k)}\leftarrow\frac{1}{Q(c^{(k)},\lambda_2^{(k)})}\cdot\exp\left(-\lambda_2^{(k)}\left((I_m\otimes e_s)^\top(f(t,c^{(k)})-y)^2\right)\right)$;
  $W^{(k)}=\operatorname{diag}(w^{(k)})$;
until $(||c^{(k)}-c^{(k-1)}||<tol)$ & $(|\lambda_2^{(k)}-\lambda_2^{(k-1)}|<tol)$ & $(||w^{(k)}-w^{(k-1)}||<tol)$;

In order to improve the convergence properties of the nonlinear scheme, we employ a continuation technique on $\overline{E^2}$. In more detail, we define a sequence of increasing reduction factors

$$1 = r_0 < r_1 < r_2 < \cdots < r_N = \frac{\overline{E^2}_{\rm uw}}{\overline{E^2}}$$

and the corresponding sequence of mean squared errors

$$\overline{E^2_j} = \frac{1}{r_j}\overline{E^2}_{\rm uw}, \quad j = 0, \dots, N, \tag{25}$$

so that $\overline{E^2_0} = \overline{E^2}_{\rm uw}$ and $\overline{E^2_N} = \overline{E^2}$. Then, for $j = 0, \dots, N$, we perform lines 2–5 of Algorithm 1 taking care that the output quantities $c^{(k)}$, $\lambda_2^{(k)}$, $W^{(k)}$ obtained at step j are used as input parameters for the subsequent step $j+1$.

A further relevant motivation for employing such a continuation technique is that it generates a discrete family of homotopic curves, parametrized by $\overline{E^2_j}$, admitting the OLS and the MEWLS solutions as initial and final configurations respectively. Each element in this family brings a specific weight distribution (and entropy value) and acts as a starting guess for the subsequent approximation curve. Therefore, the overall procedure can be interpreted as an improvement on the OLS approximation in that, by reducing the mean squared error progressively, it smoothly deforms the initial shape of the spline curve to get rid of outliers. An illustration is provided in the first example of the next section.

Finally, it is worth noticing that the resulting weights may be exploited for classification purposes. Indeed, the original data set D may be split in two disjoint subsets: $D = D_1 \cup D_2$, where D_1 contains the inliers while D_2 identifies the outliers. To this end, given a small enough tolerance tol, one can set, for example,

$$D_2 = \{(x_i, y_i) \in D \mid w_i < tol \cdot \max_j w_j\}, \qquad D_1 = D - D_2. \tag{26}$$

4. Numerical illustrations

To showcase the potential of the MEWLS spline approximation, we present three numerical experiments using synthetic data points. The first experiment focuses on a spline function fitting problem, aiming at elucidating the continuation technique (25) and the use of (26) for the automatic detection of outliers. The second and third examples involve approximating a set of data points with a spline curve in the plane and in 3D space, respectively. In the final example, we compare our method with other established smoothing techniques on a progressively challenging test problem.[4]

All the numerical tests have been implemented in Matlab (R2023a) on a 3.6 GHz Intel I9 core computer with 32 GB of memory. References to colors have been included for the online version of the manuscript.

4.1. Example 1

We consider a dataset comprising 44 points in the square $[0,1]\times[0,1]$, out of which 32 closely follow a given profile, while the remaining 12 consistently deviate from it. To fit the data, we employ a spline of degree $d = 2$, defined on a regular and uniform knot sequence consisting of $n = 17$ control points, covering the interval $[0,1]$.

In the top-left picture of Fig. 1, we observe the data set along with the ordinary least squares approximation. We see that the OLS spline approximation fails to accurately reproduce the correct profile due to the strong influence of the 12 anomalous points. Therefore, we aim to improve the approximation by decreasing the weighted mean squared error while utilizing the maximal-entropy argument to make an optimal weights selection. To this end, we consider a sequence of reduction factors distributed over the interval $[1, 500]$. For graphical clarity, we set $N = 50$ in (25) to mimic the behavior of formula (13), where the variable r continuously varies within the specified interval. Algorithm 1 generates a sequence of 50 homotopic functions with parameter $r \in [1, 500]$.

The top-right picture of Fig. 1 displays two such functions, one corresponding to $r = 2$ ($\overline{E^2} = \overline{E^2}_{\rm uw}/2$, solid line), and the other to $r = 4$ ($\overline{E^2} = \overline{E^2}_{\rm uw}/4$, dashed-line). As the reduction factor r increases, the maximum entropy principle deforms the shape of the original OLS solution by adjusting the weights to ensure that the maximum number of points contribute while still adhering to the mean squared error constraint.

In the bottom-left picture of Fig. 1 we can see the final shape of the approximating spline, corresponding to $r = 500$ ($\overline{E^2} = \overline{E^2}_{\rm uw}/500$). We can see that it nicely conforms to the profile underlying the given data set. The use of formula (26) with $tol = 10^{-4}$ correctly detects 12 outliers which are surrounded by small circles in the picture.

Finally, the bottom-right picture of Fig. 1 illustrates the behavior of the entropy (12) as a function of the scaling factor r. As expected, reducing $\overline{E^2}$ results in a decrease of the entropy associated with the weight distribution. The appropriate choice of $\overline{E^2}$ depends on the context and, in particular, on the expected accuracy of the model in the absence of outliers. An automatic identification of a suitable value for $\overline{E^2}$ may be inferred by examining the rate of change in the spline approximations as the scaling factor r increases, which is closely related to the behavior of the entropy function H as a function of r. This aspect will be the subject of future research.

[4] For the sake of reproducibility, the random number generator is initialized at the beginning of each execution.

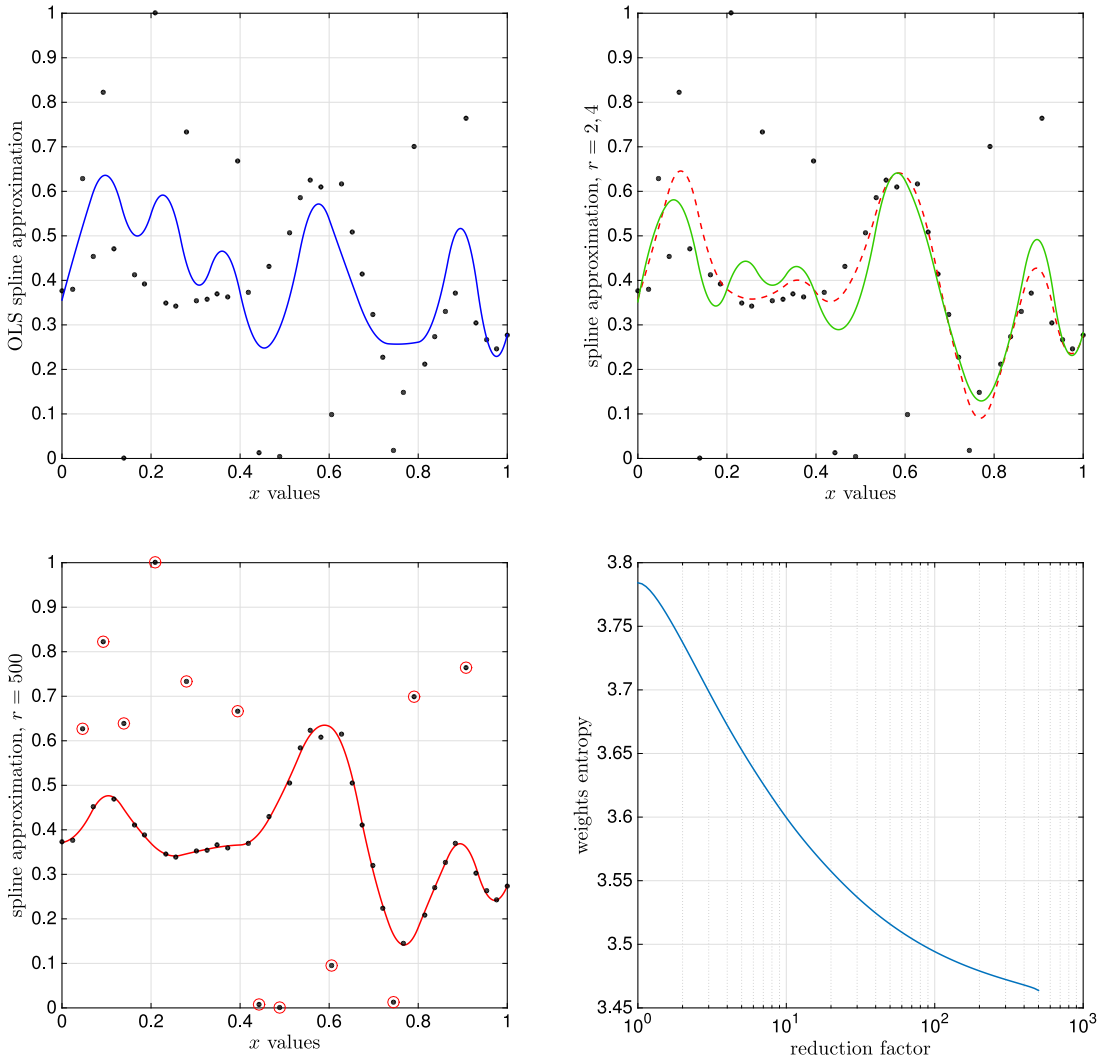

Fig. 1. Results obtained for Example 1. Top-left picture: a noisy data set revealing a pattern (dots) and its OLS spline approximation (blue line). Top-right picture: two homotopic spline functions corresponding to a reduction factor $r = 2$ (solid green line) and $r = 4$ (dashed red line). Bottom-left picture: final MEWLS spline approximation (red line) obtained by reducing the mean squared error by a factor $r = 500$. Detected outliers, identified automatically through the use of formula (26), are indicated by dots surrounded with small circles. Bottom-right picture: the entropy associated with the distribution of weights, showcased as a function of the scaling factor r.

4.2. Example 2

We address the problem of approximating the arithmetic spiral defined by the equations

$$\begin{cases} x(t) &= (a+bt)\cos(t), \\ y(t) &= (a+bt)\sin(t), \end{cases}$$

with $a = 1$, $b = 4$, $t \in [-a/b, 4\pi]$, which ensures that the spiral originates at the origin. To this end, we create a data set consisting of $N = 200$ points sampled along the spiral and then introduce random noise to 100 of them, specifically targeting the odd-numbered ones. In more detail, after setting $h = (4\pi + a/b)/(N-1)$, our data set is defined as follows:

$$\begin{cases} t_i &= -a/b + (i-1)h, & i = 1, \ldots, N, \\ (x_i, y_i) &= (x(t_i), y(t_i)), & \text{if } i \text{ is even}, \\ (x_i, y_i) &= (x(t_i) + \delta_x^{(i)}, y(t_i) + \delta_y^{(i)}), & \text{if } i \text{ is odd}, \end{cases}$$

where $\delta_x^{(i)}, \delta_y^{(i)} \in \mathcal{N}(0, \sigma^2)$ are random variables distributed normally with mean 0 and variance $\sigma^2 = 30$. Since, for the specified range of t, the spiral is entirely enclosed in the square $S = [-60, 60]^2$, for visualization clarity, we iterate the generation of values $\delta_x^{(i)}, \delta_y^{(i)}$ until (x_i, y_i) falls within S, for each odd index i.

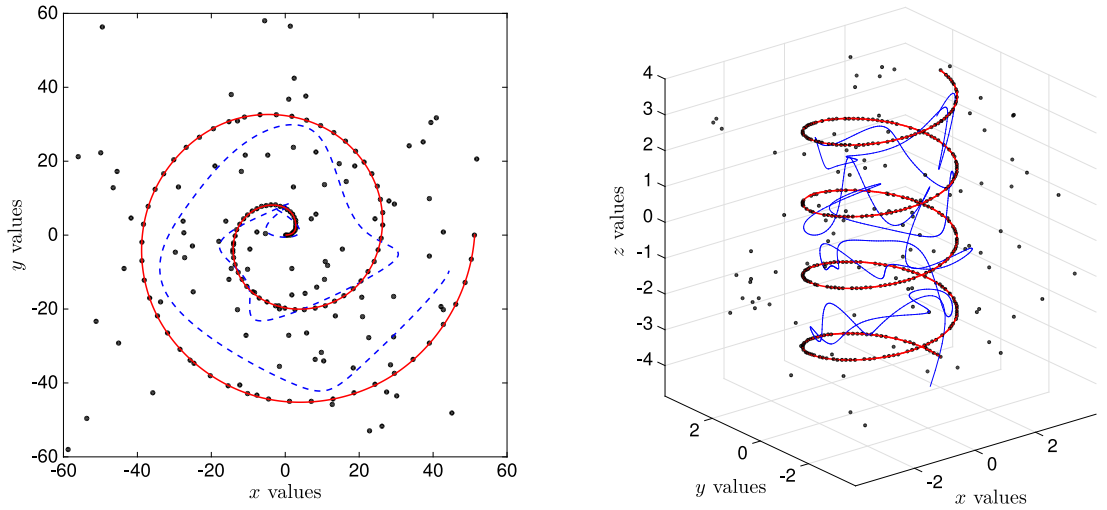

Fig. 2. Results obtained for Examples 2 and 3. Left picture: a dataset comprising 200 points, with half of them precisely aligned on an Archimedean spiral and the remainder introducing noise. Both OLS (dashed blue line) and MEWLS (solid red line) spline approximations are illustrated. Right picture: the data set consists of 400 points, with 300 of them following a circular helix pattern, while the remaining 100 contribute as noise. Both OLS (irregular blue line) and MEWLS (red line) spline approximations are displayed.

The left picture of Fig. 2 portrays the dataset $(x_i, y_i)_{i=1}^N$ along with the spline approximations of degree $d = 3$ and $n = 18$ control points using ordinary least squares (dashed line) and maximum entropy weighted least squares (solid line). Notably, while the OLS approximation struggles to capture the true spiral due to the presence of outliers, the MEWLS spline curve faithfully reproduces the unperturbed spiral $(x(t), y(t))$.

4.3. Example 3

We replicate a procedure akin to the one executed in the prior spiral example but address our attention to a circular helix defined by the equations

$$\begin{cases} x(t) &= r\cos(2\pi t), \\ y(t) &= r\sin(2\pi t), \\ z(t) &= ct, \end{cases}$$

with $r = 2, c = 1$ and $t \in [-4, 4]$, so the helix is enclosed in the cube $[-4, 4]^3$. We begin with a data set $(x_i, y_i, z_i)_{i=1}^N$ consisting of $N = 400$ points sampled along the helix but, differently to what was done in Section 4.2, we now introduce a random noise to a randomly chosen subset of these points. More precisely, we first compute a subset Ω obtained by randomly extracting $M = 100$ points from the set of indices $\{1, 2, \ldots, N\}$. Then, after setting $h = 8/(N-1)$, we define

$$\begin{cases} t_i &= -4 + (i-1)h, & i = 1, \ldots, N, \\ (x_i, y_i, z_i) &= (x(t_i), y(t_i), z(t_i)), & \text{if } i \notin \Omega, \\ (x_i, y_i, z_i) &= (x(t_i) + \delta_x^{(i)}, y(t_i) + \delta_y^{(i)}, z(t_i) + \delta_z^{(i)}), & \text{if } i \in \Omega. \end{cases}$$

Here, $\delta_x^{(i)}, \delta_y^{(i)}, \delta_z^{(i)} \in \mathcal{N}(0, 20)$ represent random variables drawn from a normal distribution with mean 0 and variance $\sigma^2 = 20$. Again, for visualization clarity, for each odd index i we iterate the generation of the perturbation values $\delta_x^{(i)}, \delta_y^{(i)}, \delta_z^{(i)}$ until (x_i, y_i, z_i) falls within the cube $S = [-4, 4]^3$. The right picture of Fig. 2 displays the dataset $(x_i, y_i, z_i)_{i=1}^N$ along with the spline approximations of degree $d = 3$ and $n = 50$ control points using ordinary least squares (irregular solid line) and maximum entropy weighted least squares (helix-shaped solid line). Again the MEWLS spline curve faithfully reproduces the shape of the original helix.

4.4. Example 4

The last illustrative example aims at comparing the MEWLS spline approximation technique with other well known smoothing methodologies accessible in Matlab through the *Curve fitting* and *Computer Vision* toolboxes. The following algorithms will be considered:

- The M-estimator Sample Consensus MSAC algorithm, a variant of the Random Sample Consensus (RANSAC) algorithm, implemented in the Matlab function `RANSAC`. Widely used for its robustness, this algorithm employs an iterative procedure to automatically discern the inlier and outlier sets. It is probabilistic in nature: at each iteration a small subset is randomly

extracted from the data set and used to estimate the parameters of the selected fitting model — specifically, a spline with the same degree and control points used in the MEWLS approximation algorithm. Subsequently, every element in the entire dataset is examined and classified as an inlier or outlier based on whether its distance from the current fitted model is smaller or larger than a given error threshold. At the next iteration a new model is fitted and compared with the previous one: the model that identifies a greater set of inliers (the consensus set) is retained while the other is discarded. This iterative process is repeated a sufficient number of times to achieve a specified confidence level, representing the probability that the final solution detects a consensus set of maximum cardinality.

- The cubic smoothing spline, a model resulting from the introduction of a penalty term in the cost function to cope with noisy observations. The cubic smoothing spline $s(x)$ approximation with uniform weights, available in Matlab through the function `FIT` with input parameter `''smoothingspline''` minimizes

$$p\sum_{i=1}^{n}(y_i - s(t_i))^2 + (1-p)\int\left(\frac{\mathrm{d}^2 s}{\mathrm{d}t^2}\right)^2 \mathrm{d}t,$$

where the smoothing parameter p ranges in $[0,1]$. Setting $p = 1$ (no smoothing term) produces the classic cubic spline interpolant, whereas $p = 0$ (no quadratic term) results in the linear least squares approximation. The parameter p may be either automatic or manually selected. In our experiment, we will explore both options.

- the robust local regression method *rlowess*, accessible in Matlab through the function `SMOOTH` with input parameter `''rlowess''`. According to this strategy, to each point (t_i, y_i) in the dataset, a weight w_i is assigned through a two-step procedure that iterates five times. In the initial step, weights w_i are assigned to points whose predictor values fall within a user-specified span. This step involves determining a local weighted least-squares regression using a first-degree polynomial. Subsequently, the second step aims at mitigating the distortion effect caused by outliers. This is achieved by re-evaluating the weights based on the residuals r_i resulting from the regression procedure.
- the robust local regression method *rloess*, available in Matlab through the function `SMOOTH` with input parameter `''rloess''`. It employs a procedure similar to the *rlowess* method, with the difference that it utilizes a quadratic polynomial for the local weighted least-squares regression.

The test problem is prepared as follows. We begin with sampling the sigmoid function $f(t) = 1/(1+e^{-t})$ over $N = 1000$ equidistant points covering the interval $[a,b] = [-10,10]$. Then, as in the previous example, a subset Ω is created by randomly extracting M points from the set of indices $\{1,2,\ldots,N\}$. We now introduce two sources of perturbation, each acting on one of the two partitions induced by Ω, according to the following rule:

$$\begin{cases} t_i &= a + (i-1)h, & i = 1,\ldots,N,\\ y_i &= f(t_i) + \delta^{(i)}, & \text{if } i \notin \Omega,\\ y_i &= f(t_i) + \varepsilon^{(i)}, & \text{if } i \in \Omega, \end{cases} \tag{27}$$

where $h = (b-a)/(N-1)$, $\delta^{(i)}$ is a random variable drawn from a normal distribution with mean 0 and standard deviation growing linearly with t as $\sigma(t) = (t-a)/500$, $t \in [a,b]$, while $\varepsilon^{(i)}$ is a random variable drawn from a uniform distribution in the interval $[0, f(t_i)]$.

A remark is in order to clarify the nature of the two kinds of perturbations and their effect on the final shape of the dataset. Each point in the dataset is affected by an error. Points that lie outside the set identified by Ω are subjected to Gaussian noise whose variance varies with time. In contrast, most of the points identified by the set Ω may be labeled as outliers, as their locations are determined by a uniform distribution and can significantly deviate from the pattern defined by the sigmoid function. The resulting dataset mirrors the common experience with data coming from the observation of real phenomena: a noisy pattern describing the relationship between the interested physical quantity and an explanatory variable, along with the possible presence of data that substantially deviate from the underlying pattern.

The goal is to evaluate how the performance of our entropy-based methodology compares with other well-known robust strategies capable of handling very noisy datasets. The possibility of increasing the value of M (and thus the number of outliers), combined with the fact that all outliers lie on one side of the graph of the function $f(t)$ (thus avoiding a compensation effect), allows us to progressively make the challenge of reproducing the correct sigmoid function more and more difficult. Indeed, we have selected three levels of difficulty:

- *normal*: $M = 100$, where about 10% of data are outliers;
- *hard*: $M = 500$, representing about 50% of data as outliers;
- *extreme*: $M = 800$, with a proportion of outliers reaching about 80%.

In all the experiments the spline embedded in the MEWLS and RANSAC models has degree $d = 3$ and $n = 10$ control points. The normal difficulty level has been considered to start from a scenario where all the considered methods give satisfactory results, which are displayed in Fig. 3. In the top-left picture, we observe the data set (black dots) along with the sigmoid function (green dashed line), the ordinary least squares approximation (blue dotted line), and the MEWLS spline approximation (red solid line). The top-right picture shows the same data except for the red solid line which now corresponds to the output of the RANSAC algorithm with the confidence parameter set to its default value 99%. Moving to the bottom-left picture, the results of the smoothing spline approximation are presented: the blue solid line corresponds to the automatic selection of the smoothing parameter, namely

Table 1
Root mean squared errors and coefficients of determination produced by the six strategies employed in Example 4. Negative R^2 values, symptomatic of very poor performance, and are indicated by an asterisk.

Difficulty level	GOF	OLS	MEWLS	RANSAC	s-spline	RLOWESS	RLOESS
normal	RMSE(f)	8.0e−2	4.1e−3	5.4e−3	7.9e−2	5.6e−3	4.9e−3
	RMSE($\tilde{f}$)	8.3e−2	2.3e−2	2.4e−2	8.2e−2	2.4e−2	2.3e−2
	$R^2(f)$	0.9681	0.9999	0.9998	0.9684	0.9998	0.9999
	$R^2(\tilde{f})$	0.9659	0.9973	0.9972	0.9663	0.9972	0.9973
hard	RMSE(f)	3.9e−2	6.8e−3	2.3e−2	3.9e−1	2.4e−1	2.5e−1
	RMSE($\tilde{f}$)	3.9e−1	2.4e−2	3.2e−2	3.9e−1	2.4e−1	2.5e−1
	$R^2(f)$	0.2550	0.9998	0.9975	0.2535	0.7030	0.6921
	$R^2(\tilde{f})$	0.2555	0.9971	0.9948	0.2539	0.7019	0.6909
extreme	RMSE(f)	6.1e−1	1.2e−2	2.5e−1	6.1e−1	5.9e−1	5.9e−1
	RMSE($\tilde{f}$)	6.1e−1	2.6e−2	2.5e−1	6.1e−1	5.9e−1	5.9e−1
	$R^2(f)$	*	0.9992	0.6879	*	*	*
	$R^2(\tilde{f})$	*	0.9966	0.6886	*	*	*

$p = 0.99999910844$, while the red dashed line is obtained by manually setting $p = 0.15$. Finally, the bottom-right picture exhibits overlapping curves obtained from the rlowess (red solid line) and rloess (blue dashed line) techniques with span parameter set to 10%.

Overall, the proportion of outliers is small enough that all methods, with appropriate parameters selection, perform quite well. In fact, in all cases, except perhaps for the smoothing spline, the approximations precisely capture the shape of the sigmoid function.

The four images in Fig. 4 show the outcomes achieved when the difficulty level is set to 'hard', meaning that approximately half of the points in the dataset are outliers.[5] It is evident that only the MEWLS and RANSAC algorithms manage to accurately replicate the correct approximation. In contrast, the approximations generated by the smoothing spline, rlowess, and rloess methods fail to successfully capture the correct sigmoid profile.

Encouraged by the robust performance of the MEWLS and RANSAC algorithms, the survivors of the hard difficulty challenge, we further increase the difficulty to an 'extreme' level, with about 80% of the points playing the role of outliers. Fig. 5 shows the results for the spline approximations generated by the MEWLS (left plot) and RANSAC (right plot) methods. Remarkably, the RANSAC algorithm requires approximately 54.4 s to produce an (incorrect) fitting model consistent with the default 99% confidence parameter, while the MEWLS algorithm achieves the correct profile in just 0.46 s.

To translate the graphical output in numbers and assess the goodness of fit (GOF) of the MEWLS approach in accurately reproducing the behavior of the sigmoid function, we consider a couple of statistics, namely the root mean squared error (RMSE) and the coefficient of determination R^2. It is worth noticing that, in the presence of outliers, these metrics might yield misleading results when applied to the entire dataset, as is typical in classical settings (see for example [23]). We circumvent this issue by comparing the values predicted by the output models with respect to two distinct reference datasets: the clean signal $\{(t_i, f(t_i))\}_{i=1}^N$ (the original function our models are asked to mimic), and the Gaussian-perturbed signal $\{(t_i, \tilde{f}(t_i))\}_{i=1}^N$, where $\tilde{f}(t_i) = f(t_i) + \delta^{(i)}$, $i = 1, \dots, N$, with $\delta^{(i)}$ defined as in (27) across all the predictors (without forced outliers).[6]

Table 1 summarizes the results for the three examined difficulty levels. The "s-spline" column refers to the outcomes produced by the smoothing-spline approximation with the smoothing parameter manually set to $p = 0.15$ (the results corresponding to the automatic selection of the smoothing parameter are omitted due to poor performance). We observe that the MEWLS spline approximation exhibits the best results, especially in relation to the extreme difficulty case (an asterisk indicates a negative R^2 value).

The results obtained in this and previous examples underscore the effectiveness of MEWLS in successfully detecting and eliminating outliers from highly noisy datasets. Further instances based on real data are illustrated in the next section.

5. A few applications to real data

5.1. Approximating the main sequence in a Hertzsprung–Russell diagram

The Hertzsprung–Russell (HR) diagram is a graphical representation of stars, mapping the correlation between their absolute magnitudes or luminosities versus their color indices or temperatures, allowing astronomers to discern distinct patterns in stellar evolution [24,25].

The absolute magnitude of a star is a measure of its intrinsic brightness or luminosity, unaltered by its distance from Earth. It is the apparent magnitude (brightness as seen from Earth) that a star would have if it were located at a standard distance of 10 parsecs (about 32.6 light-years) away. Essentially, the absolute magnitude allows astronomers to compare the luminosities of stars irrespective of their varying distances from us.

[5] The interpretation of the four pictures remains consistent with that of Fig. 3.

[6] The random number generator is not initialized here, in order to generate different values from those used during the training phase.

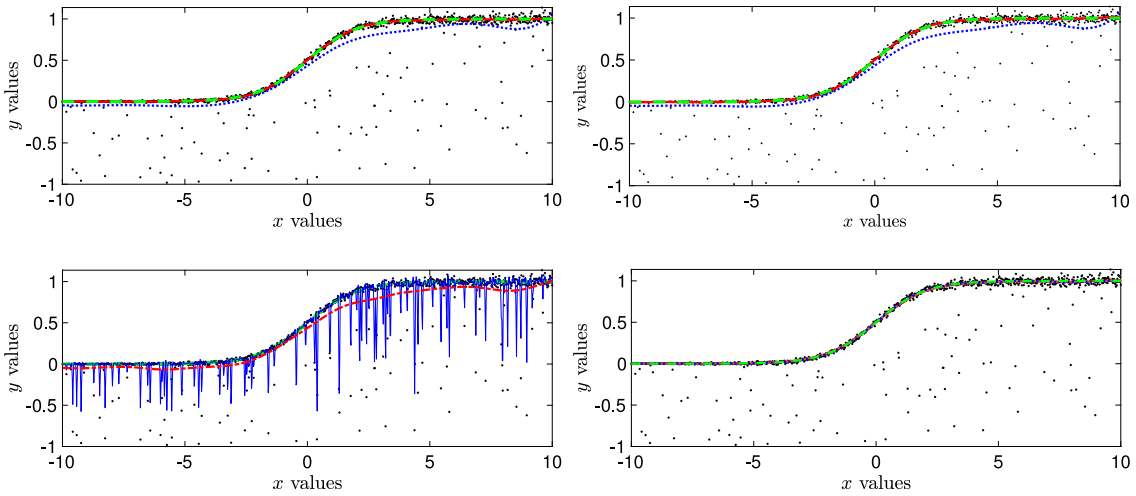

Fig. 3. Results for Example 4, with $M = 100$ (normal difficulty level). The data set (black dots) reproduces a sigmoidal pattern (green dashed line). Top-left picture: OLS spline approximation (blue dotted line) and the MEWLS spline approximation (red solid line). Top-right picture: OLS spline approximation (blue dotted line) and spline approximation obtained by the RANSAC algorithm (red solid line). Bottom-left picture: Smoothing spline approximations obtained by the automatic selection of the smoothing parameter (blue solid line) and smoothing parameter $p = 0.15$ (red dashed line). Bottom-right picture: Approximations produced by rlowess (blue dotted line) and rloess (red solid line) methods with span parameter set to 0.1.

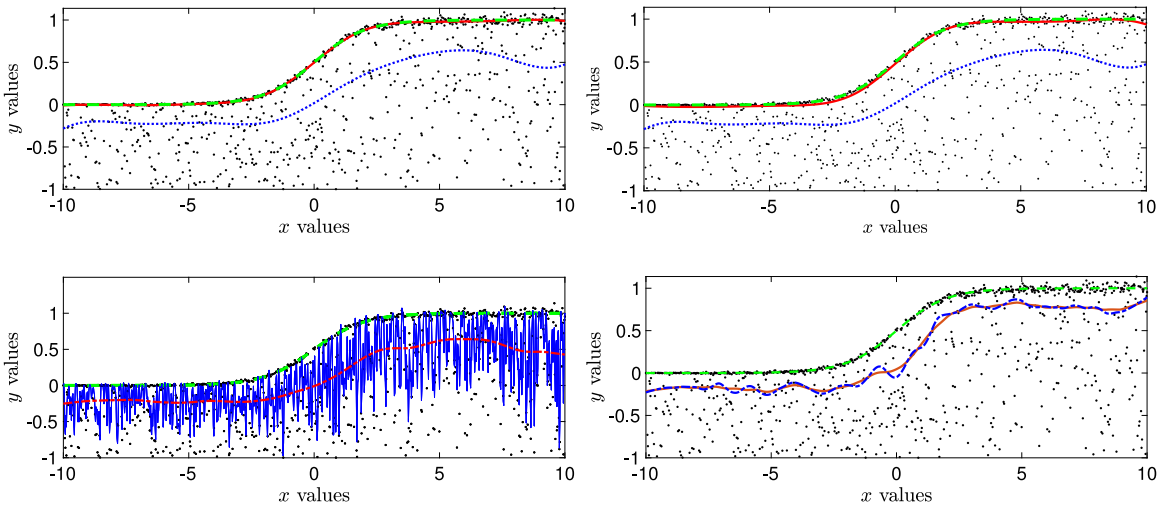

Fig. 4. Results for Example 4, with $M = 500$ (hard difficulty level). The interpretation of the four pictures is the same as in Fig. 3.

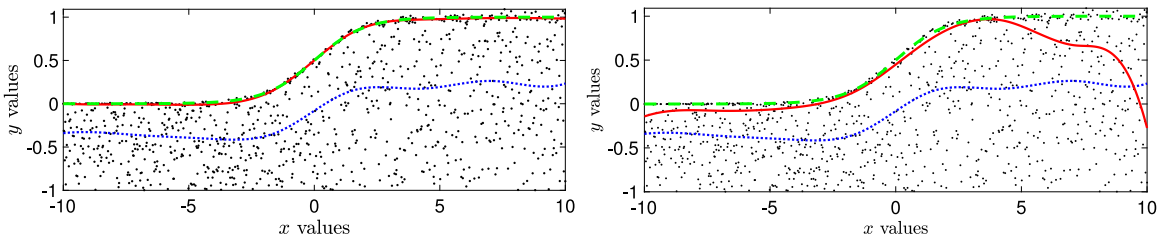

Fig. 5. Results for Example 4, with $M = 800$ (extreme difficulty level). The sigmoid function (green dashed line), OLS spline approximation (blue dotted line), MEWLS spline approximation (left picture, red solid line) and RANSAC spline approximation (right picture, red solid line).

The B-V color index is a parameter that characterizes a star's color and temperature. It is the difference between the star's apparent magnitudes in the blue (B) and visual (V) parts of the electromagnetic spectrum. Blue stars have negative B-V values,

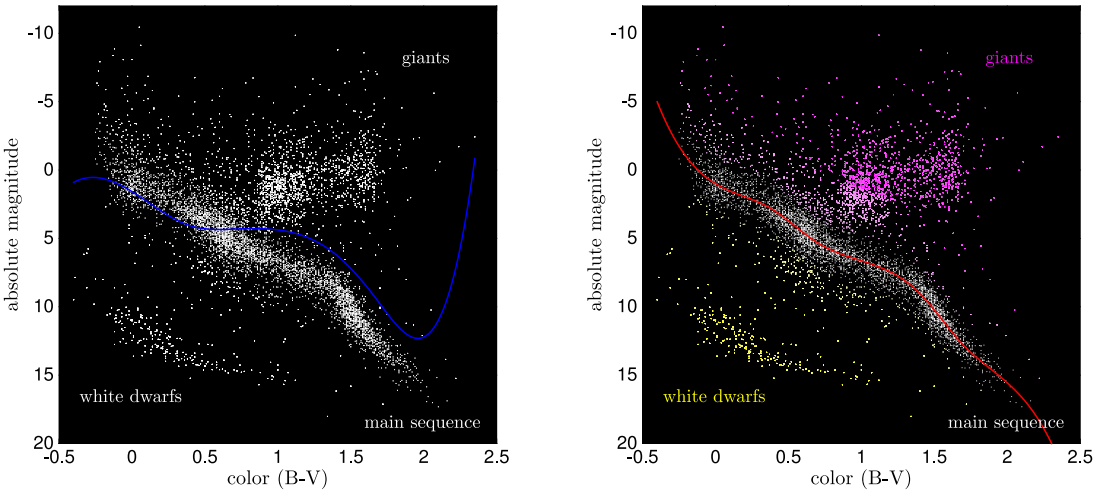

Fig. 6. Hertsprung-Russel diagrams of the Yale dataset. Left picture: ordinary least squares spline approximation (blue line). Right picture: maximal-entropy least squares spline approximation (red line). The intensity of magenta and yellow colors is inversely proportional to the weight associated with each data point.

while redder stars have positive values. This index is crucial in categorizing stars by their spectral types, indicating whether a star is hotter (blue) or cooler (red).

Together, the absolute magnitude and B-V color index are vital tools in understanding stars' properties, evolutionary stages, and positions within the Hertzsprung–Russell diagram. As an example, the left picture of Fig. 6 shows the HR diagram for the Yale Trigonometric Parallax Dataset [26] comprising more than 6000 catalogued stars. This astronomical resource provides measurements of stellar distances using the trigonometric parallax method, a technique employed to determine the distance to a star by measuring its apparent shift in position against more distant background stars as the Earth orbits the Sun. Besides observed parallaxes (in arcsec), the Yale catalogue also includes the B-V color index and the apparent V magnitude. The absolute magnitude is then obtained by means of the formula

$$\text{absolute magnitude} = \text{apparent V magnitude} + 5(\log_{10}(\text{observed parallax}) + 1).$$

At its core, the diagram features a continuous and well-defined band known as the main sequence. This band comprises the vast majority of genuine stars in the cosmos, including our own Sun with an absolute magnitude of 4.8 and a B-V color index of 0.66.

Located in the lower-left portion of the diagram are the white dwarfs, while the upper part accommodates the subgiants, giants, and supergiants. This layout visually captures the diverse stages of stellar evolution with the white dwarfs representing stars in their final stages of evolution.

One of the diagram's remarkable applications is in determining the distance between Earth and distant celestial objects like star clusters or galaxies.

In this example, our aim is to accurately approximate the main sequence's shape using an appropriate spline curve and further categorize stars through color assignments. To achieve this, we employed a spline of degree $d = 3$ along with a regular knot sequence

$$t = [0, 0, 0, 0, 0.286, 0.397, 0.658, 0.757, 1, 1, 1, 1].$$

The left picture in Fig. 6 displays the outcome of the ordinary least squares approximation (indicated by the blue line in the color image). This method evidently fails to accurately replicate the main sequence's distinctive form due to the presence of giants and white dwarfs. Conversely, the maximal-entropy least squares approximation, indicated by the red line in the right picture of Fig. 6, successfully captures the main sequence's true shape. By determining the distribution of weights based on the entropy-driven procedure, we assigned distinct color gradients to each star. This color differentiation effectively highlights the discrepancies between white dwarfs/giant stars and those belonging to the main sequence. As the corresponding weights decrease, the intensity of magenta and yellow pixels progressively intensifies. This approach not only improves the accuracy of the main sequence representation but also facilitates the identification of stars that deviate from its expected characteristics.

5.2. Detecting train rails in a railway infrastructure and surrounding environment

In the present example, we delve into a segmentation task performed on a point cloud that portrays a railway environment, captured using a terrestrial laser scanning system. An instance of such a scenario is presented in Fig. 7, which will serve as the subject of our examination. Here, we observe a curved railway emerging from a tunnel, enveloped by dense vegetation. Our aim revolves around identifying the train rails within this scenario and approximating their shape using a suitable spline curve. Conducting such an analysis can yield valuable insights into the transportation system and aid in identifying potential issues that could impact its operational effectiveness (see [27] and reference therein).

Fig. 7. A visual representation of a 3D point cloud showcasing a curved section of a railway emerging from a tunnel, with the surrounding vegetation captured in the scene.

It is worth underscoring that the essence of this example lies in testing the entropy-based approach on a highly noisy dataset, where the set D_1 of inliers is significantly dwarfed by the set D_2 of outliers. As a result, the technique showcased in this example serves as a proof of concept rather than a definitive solution for the intended problem (for a more effective identification of the rails, refer to works such as [28–30]).

A point cloud is a data set that realizes a digital representation of a physical environment or object in a three-dimensional space. It is arranged in a structured array housing fields that store various attributes for each point within the cloud. These attributes encompass 3D coordinates, distance ranges, color information, intensity measurements, and potentially other geometric or spectral data. We will utilize the intensity parameter, a measure of the reflectivity of the material of the object containing the sample point, to identify reflective elements like train rails.

Within the segmentation procedure, the intensity field frequently comes into play for the purpose of condensing the initial array of data points into a more fitting subset of points pertinent to the analysis. In fact, noteworthy structures, including train rails and overhead wires, exhibit resemblances in their intensity attributes. This correspondence arises from the inherent connection between a surface's reflective characteristics and its constituent material. For instance, train rails are predominantly composed of steel, leading to nearly uniform intensity readings from the laser sensor along the rail's length. By resorting on the intensity parameter as a filtering criterion, we can effectively discern the majority of points situated on the rails.

Building upon the analysis conducted in [30] for a point cloud of similar nature, our approach to reduce the size of the original point cloud, while retaining the majority of rail points, involves extracting those with intensity values not exceeding 65. Additionally, due to the level nature of the terrain under consideration, we omit the vertical component of the points and instead focus on a two-dimensional projection of the filtered point cloud. This projection is illustrated in the leftmost image of Fig. 8 and forms a data set comprising 304911 points. The lower-right section of the image corresponds to the segment of the rails situated within the tunnel. This region exhibits a much cleaner appearance compared to the area outside the tunnel. Indeed, in the external environment, a considerable number of points associated with vegetation are regrettably retained even after the filtering procedure. This introduces a notable degree of noise into the data.

The right image in Fig. 8 displays the ordinary least squares spline approximation curve (solid blue line). By referring to Eqs. (1)–(2), this curve is obtained through a spline of degree $d = 2$ and $n = 15$, utilizing a uniform $(d+1)$-regular knots distribution. Evidently, the OLS approximation does not deviate that much from the shape traced by the rail tracks, making it a suitable initial estimate within Algorithm 1 for computing the maximal-entropy weighted least squares spline approximation curve.

This MEWLS curve is depicted in the same graph as a dashed red line. It is clear that the MEWLS spline closely captures the profile of the upper rail, demonstrating a very high accuracy. An inspection of the weights through formula (26) reveals that, in this specific example, the number of outliers exceeds the number of inliers by more than six times.

A comparable process can be subsequently applied to acquire the approximation for the lower rail. This involves eliminating the points related to the upper rail from the dataset and then performing Algorithm 1 again (we omit to display this latter approximation for visual clarity).

In conclusion, the MEWLS approach effectively enhances the accuracy of the initial OLS approximation and leads to a precise parametric representation of the rails.

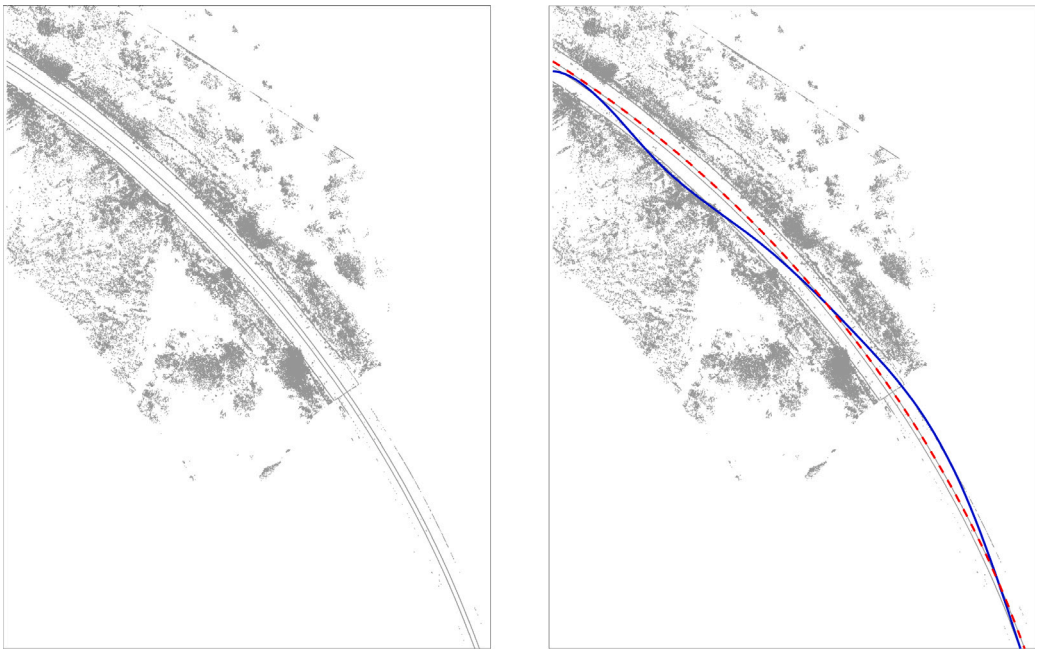

Fig. 8. Left picture: 2D projection of the filtered point cloud. The tracks are correctly represented but, unfortunately, vegetation outside the gallery introduces a relevant number of noisy points in the filtered image. Right picture: ordinary least squares spline approximation (solid blue line) and maximal-entropy least squares spline approximation (dashed red line).

5.3. Detecting and scoring outliers in an environmental data set

The final test case is drawn from a study in [6] and explores an environmental dataset accessible through the R-package *openair* [31]. This dataset encompasses hourly readings of wind speed, wind direction, and concentrations of pollutants such as NO_x, NO_2, O_3, PM_{10}, SO_2, CO, and PM_{25} recorded at Marylebone (London) spanning from January 1, 1998, to June 23, 2005. For comparison purposes, we conform to the choice in [6] and focus on a specific subset of this dataset, only comprising the O_3 concentrations during December 2002. This particular segment encompasses a total of 744 observations, while also featuring several instances of missing data points.

The dots depicted in Fig. 9 provide a visual representation of the O_3 concentrations, measured in parts per billion (ppb), over the specified time frame. To approximate this univariate time series, we employ a spline function with degree $d = 3$, defined on a uniform $(d + 1)$-regular knots distribution. In order to capture the erratic nature of the data, we opt for a number of coefficients n equal to half the data points' count. Fig. 9 only displays the MEWLS approximation (red solid line).

In contrast to the approach adopted in prior examples, our strategy for obtaining the approximating spline varies here. Rather than predefining the reduction factor, we pursue a distinct perspective. Specifically, we establish the number of outlier candidates, denoted as N, and iteratively reduce the $\overline{E^2}$ value until N data points are encompassed within the outlier set D_2. This methodology introduces a natural ranking within D_2, assigning scores to each prospective outlier. This is readily accomplished using (26), where the ith point entering D_2 receives a score of i. In Fig. 9, outliers are denoted by points enclosed in green circles, each indicating the corresponding score.

The outcomes obtained align with those presented in [6], particularly those based upon the extreme value theory. This systematic scoring approach has the potential to streamline the decision-making process, aiding specialists in identifying the data points that merit closer investigation or intervention.

6. Conclusions

In real-world scenarios, data quality directly impacts the performance of subsequent analytical processes, so that the importance of effective preprocessing techniques and robust fitting procedures have become increasingly evident.

In this context, we have introduced an entropy-based weighting methodology for determining spline approximations of multivariate time series. In contrast to the ordinary least squares approach, which displays sensitivity to corrupted data, the MEWLS spline approximation effectively mitigates the impact of outliers and noise even when handling large and highly noisy datasets. Its ability to accurately extract meaningful information from noisy backgrounds has been illustrated through various synthetic and real-world examples.

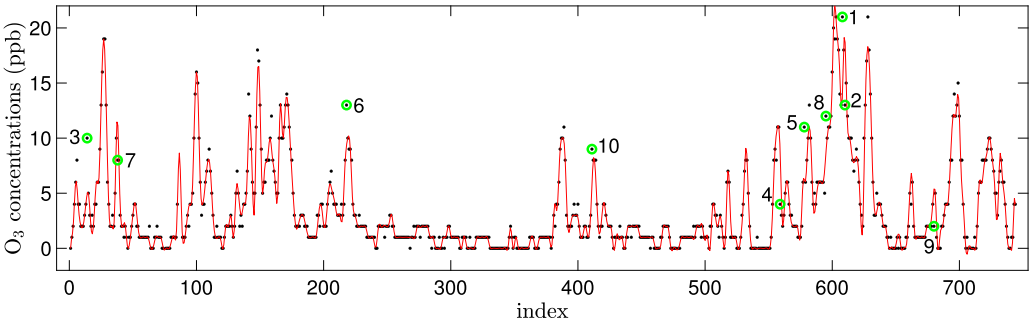

Fig. 9. Dots: Hourly O_3 concentrations recorded at Marylebone during December 2002 (taken from the R package *openair*). Solid red line: maximal entropy least squares spline approximation. Dots surrounded by green circles identify the first ten outliers detected by the procedure.

One limitation when compared to the OLS approach is that, even for linear models, the resulting algebraic system becomes nonlinear and its solution requires the implementation of an appropriate iterative scheme. In this regard, the OLS solution can serve as an initial estimate. The numerical illustrations underscore that the MEWLS solution significantly outperforms the classical OLS procedure. Nonetheless, the efficient resolution of this nonlinear system warrants dedicated investigation and will be a focus of future research.

Finally, it is worth highlighting that the entropy-based approach employed to address the issue of erroneous data is versatile and can be exported to various contexts where the primary objective is fitting a given dataset using an appropriate model. The key assumption enabling the application of this approach is that the cost function to be minimized can be expressed as a combination, for example an average, of contributions from individual predictor-outcome pairs in the dataset. Therefore, potential avenues for future investigation encompass, but are not limited to, linear fitting with different basis functions, multivariate regression, nonlinear fitting, as well as training machine learning and deep learning models.

CRediT authorship contribution statement

Luigi Brugnano: Conceptualization, Investigation, Methodology, Visualization, Writing – review. **Domenico Giordano:** Conceptualization, Investigation, Methodology, Visualization, Writing – review. **Felice Iavernaro:** Conceptualization, Data curation, Validation, Investigation, Writing – review & editing, Supervision, Software. **Giorgia Rubino:** Conceptualization, Data curation, Validation, Investigation, Writing – review & editing, Software, Supervision.

Data availability

The authors do not have permission to share data.

Acknowledgments

The authors wish to thank the two reviewers for providing valuable comments and suggestions that surely contributed to enhancing the final shape of the manuscript.

Felice Iavernaro acknowledges the contribution of the National Recovery and Resilience Plan, Mission 4 Component 2 - Investment 1.4 - NATIONAL CENTER FOR HPC, BIG DATA AND QUANTUM COMPUTING - Spoke 5 - Environmental and Natural Disasters, under the NRRP MUR program funded by the European Union - NextGenerationEU - (CUP H93C22000450007).

Luigi Brugnano and Felice Iavernaro thank the GNCS for its valuable support under the INDAM-GNCS project CUP_E55F22000270001.

References

[1] X. Zhu, X. Wu, Q. Chen, Eliminating class noise in large datasets, in: ICML, 2003, pp. 920–927.
[2] C.M. Teng, Correcting noisy data, in: ICML, 1999, pp. 239–248.
[3] D. Gamberger, N. Lavrac, S. Dzeroski, Noise detection and elimination in data preprocessing: Experiments in medical domains, Appl. Artif. Intell. 14 (2) (2000) 205–223.
[4] A. Zimek, P. Filzmoser, There and back again: Outlier detection between statistical reasoning and data mining algorithms, Wiley Interdiscipl. Rev.: Data Min. Knowl. Discov. 8 (6) (2018) e1280.
[5] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. 41 (3) (2009) 1–58.
[6] M. Čampulová, R. Čampula, J. Holešovský, An R package for identification of outliers in environmental time series data, Environ. Model. Softw. 155 (2022) 105435.
[7] A. Farhangi, J. Bian, A. Huang, H. Xiong, J. Wang, Z. Guo, AA-forecast: Anomaly-aware forecast for extreme events, Data Min. Knowl. Discov. 37 (3) (2023) 1209–1229.
[8] D. Giordano, F. Iavernaro, Maximal-entropy driven determination of weights in least-square approximation, Math. Methods Appl. Sci. 44 (2021) 6448–6461.

[9] A. Falini, F. Mazzia, C. Tamborrino, Spline based Hermite quasi-interpolation for univariate time series, Discrete Contin. Dyn. Syst. - Series S 15 (12) (2022) 3667–3688.
[10] A. Raffo, S. Biasotti, Weighted quasi-interpolant spline approximations: Properties and applications, Numer. Algorithms 87 (2021) 819–847.
[11] T. Hastie, R. Tibshirani, J.H. Friedman, The Elements of Statistical Learning, Data Mining, Inference, and Prediction, second ed., in: Springer Series in Statistics, 2009.
[12] G. Wahba, Spline Models for Observational Data, in: CBMS-NSF Regional Conference Series in Applied Mathematics, Series Number 59, Society for Industrial and Applied Mathematics, Philadelphia, 1990.
[13] R. Andersen, Modern methods for robust regression, in: Sage University Paper Series on Quantitative Applications in the Social Sciences, 2008, pp. 07–152.
[14] P.J. Rousseeuw, A.M. Leroy, Robust Regression and Outlier Detection, Wiley, 2003.
[15] T. Strutz, Data Fitting and Uncertainty (a Practical Introduction To Weighted Least Squares and beyond), Springer Vieweg, 2016.
[16] C. Yu, W. Yao, Robust linear regression: A review and comparison, Comm. Statist. Simulation Comput. 46 (8) (2017).
[17] W.S. Cleveland, Robust locally weighted regression and smoothing scatterplots, J. Amer. Statist. Assoc. 74 (368) (1979) 829–836.
[18] W.S. Cleveland, LOWESS: A program for smoothing scatterplots by robust locally weighted regression, Amer. Statist. 35 (1) (1981) 54.
[19] C. Loader, Local Regression and Likelihood, Springer, New York, NY, 1999.
[20] M.A. Fischler, R.C. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395.
[21] E.T. Jaynes, Foundations of probability theory and statistical mechanics, in: B. Mario (Ed.), Delaware Seminar in the Foundations of Physics, in: Studies in the Foundations Methodology and Philosophy of Science, vol. 1, Springer-Verlag, New York NY, 1967, pp. 77–101.
[22] T. Lyche, C. Manni, H. Speleers, Foundations of spline theory: B-splines, spline approximation, and hierarchical refinement, in: Lecture Notes in Mathematics, vol. 2219, 2018, pp. 1–76.
[23] O. Renaud, M.-P. Victoria-Feser, A robust coefficient of determination for regression, J. Statist. Plann. Inference 140 (2010) 1852–1862.
[24] D. Maoz, Astrophysics in a Nutshell, sexond ed., Princeton University Press, 2016.
[25] B. Carroll, D. Ostlie, An Introduction To Modern Astrophysics, second ed., Cambridge University Press, 2017.
[26] W.F. van Altena, J.T. Lee, E.D. Hoffleit, The General Catalogue of Trigonometric Parallaxes, fourth ed., Yale University Observatory, 1995, http://www.astro.yale.edu/astrom/YPC95.html.
[27] E. Che, J. Jung, M.J. Olsen, Object recognition, segmentation, and classification of mobile laser scanning point clouds: A state of the art review, Sensors 19 (4) (2019) 810.
[28] Y. Lou, T. Zhang, J. Tang, W. Song, Y. Zhang, L. Chen, A fast algorithm for rail extraction using mobile laser scanning data, Remote Sens. 10 (12) (2018) 1998.
[29] M. Arastounia, Automated recognition of railroad infrastructure in rural areas from LiDAR data, Remote Sens. 7 (11) (2015) 14916–14938.
[30] P. Amodio, M. De Giosa, F. Iavernaro, R. La Scala, A. Labianca, M. Lazzo, F. Mazzia, L. Pisani, Detection of anomalies in the proximity of a railway line: A case study, J. Comput. Math. Data Sci. 4 (2022) 100052.
[31] D.C. Carslaw, K. Ropkins, Openair – An R package for air quality data analysis, Environ. Model. Software 27-28 (2012) 52–61.