2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

A Recursive Implementation of Sparse Regression

Simone Baldi, *Senior Member, IEEE* and Di Liu, *Member, IEEE*

***Abstract*— Sparse regression deals with the problem of representing a dataset using only a few non-zero basis elements. This work presents a recursive implementation of sparse regression, with the dataset being processed sequentially rather than as a batch. The algorithm, named sparse regularized fused recursive least squares (SP-RF-RLS), uses a re-weighting technique and a smooth approximation to deal with the discontinuous ℓ_0-norm and the non-differentiable ℓ_1-norm, standard norms for sparsity. Inspired by fused least absolute shrinkage and selection operator (fused-LASSO), the algorithm aims to capture structures in the locations of the non-zero elements by including a term depending on the difference between the estimated elements. Comparative experiments in both sparse and non-sparse scenarios show that SP-RF-RLS outperforms several state-of-the-art recursive algorithms.**

I. Introduction

In many applications of data science, spanning from computer vision to fault and structure detection [1]–[3], it is of interest to represent data using a combination of a few basic elements. If such few elements are able to reconstruct the original data, a sparse model is obtained, whose compact nature can help towards explainability and reduction of complexity. While the ℓ_0-norm is a typical measure of sparsity, its discontinuous nature may result in instability, and alternatives like the ℓ_1-norm can be adopted [4], [5]. The most common sparse regression algorithms are non-recursive, i.e., they process data as a batch. Big families of non-recursive sparse regression are relaxation algorithms such as basis pursuit and greedy algorithms such as matching pursuit [6], [7]. The celebrated least absolute shrinkage and selection operator (LASSO) [8], [9] belongs to the family of basis pursuit. Proposed non-parametric sparse regression methods [10]–[12] are also non-recursive.

When data are collected continuously, recursive implementations of sparse regression are more appropriate. Recursive sparse regression algorithms are inspired by adaptive filtering [13], with big families falling in the least mean square (LMS) methods and the recursive least squares (RLS) methods. Due to the challenges of dealing with the ℓ_0-norm or ℓ_1-norm in a recursive way, the first recursive sparse regression algorithms utilized the ℓ_2-norm, as in Normalized LMS [14] and Proportionate Normalized LMS [15] algorithms. The zero-attracting LMS (ZA-LMS) algorithm was one of the first dealing with the ℓ_1-norm in a recursive way [16]. Studies have shown that, thanks to the better effect of the ℓ_1-norm of inducing sparsity as compared to the ℓ_2-norm, zero-attracting methods have faster convergence and higher accuracy when the system under consideration is sparse [17]. Meanwhile, the faster convergence and higher accuracy of RLS methods as compared to LMS methods, well-known in ℓ_2 regularization [18]–[20], also applies to ℓ_1-norm: this was experienced in ℓ_1-RLS and ℓ_1-RRLS algorithms [21], [22], up to variants with different regularization and weighted terms [23]–[27]. Zero-attracting RLS (ZA-RLS) algorithms have also been derived [28] to improve ZA-LMS.

Despite their good sparsity effects, these algorithms deal with each element separately: this makes it hard to capture possible structural correlations in the location of the non-zero elements. In applications where such correlations do exist (e.g., in time series or image data with spatial or temporal structure [29]), one may end up estimating non-zero elements in wrong locations. This problem has been addressed in a non-recursive way in LASSO algorithms, leading to several variants of the so-called fused-LASSO [30], [31], where the term 'fused' refers to penalizing the differences between the estimated coefficients to make non-zero elements cluster together. Despite several non-recursive algorithms for fused sparse regression, we are not aware of recursive algorithms, which is the main contribution of this work. We achieve a recursive implementation as follows:

- To induce sparsity, we introduce appropriate weights in the cost, inspired by the re-weighting technique [23], [24]. However, we use a less restrictive approach to minimize the regression cost;
- As previously noted by the authors [32], standard re-weighting suffers from lack of differentiability that also arises in ZA-RLS algorithms [28]. We thus introduce a smooth approximation of non-differentiable terms that can be handled by the minimization.
- Structural correlations are captured by including a term depending on the difference between the estimated elements, as inspired by fused-LASSO [30], [31]. However, such term is handled recursively in the minimization, rather than as a batch.

We name the algorithm sparse regularized fused recursive least squares (SP-RF-RLS). Comparative experiments in both sparse and non-sparse scenarios show that SP-RF-RLS outperforms, in terms of sparsity and accuracy of the estimate, state-of-the-art recursive algorithms, including SP-R-RLS (without fused) by some of the authors [33].

The rest of the paper is organized as follows: Sect. II recalls basic concepts of sparse regression. The proposed SP-

This research was partly supported by Natural Science Foundation of China grants 62073074 and 62233004, by Jiangsu Provincial Scientific Research Center of Applied Mathematics grant BK20233002, by Horizon Europe Marie Sklodowska-Curie Action no. 101146446, by UKRI Research and Innovation no. EP/Z002214/1, and by European Union's Horizon 2020 R&I programme under the Marie Sklodowska-Curie grant 899987.

S. Baldi is with School of Mathematics, Southeast University, China simonebaldi@seu.edu.cn

D. Liu is with Department of Electrical and Electronic Engineering, Imperial College, London SW7 2BT, U.K. di.liu@imperial.ac.uk

RF-RLS along with its theoretical basis is in Sect. III. The algorithm is compared with existing algorithms in Sect. IV. Conclusions are in Sect. V.

II. Introduction to Sparse Regression

Given a vector $\boldsymbol{w} = [w_1\, w_2 \cdots w_M]^\top$, its ℓ_p-norm, with $p > 0$, is defined as

$$\|\boldsymbol{w}\|_p = (|w_1|^p + |w_2|^p + \cdots + |w_M|^p)^{1/p}\,, \qquad (1)$$

converging, for $p \to 0$, to the ℓ_0-norm, $\|\boldsymbol{w}\|_0 = |w_1|^0 + |w_2|^0 + \cdots + |w_M|^0$ (upon the definition $0^0 = 0$). Let $\boldsymbol{w}^* \in \mathbb{R}^M$ represent the unknown elements in a system

$$\boldsymbol{Y}(k) = \boldsymbol{X}(k)\boldsymbol{w}^* + \boldsymbol{n}(k), \qquad (2)$$

being $\boldsymbol{n}$ observation noise, and $\boldsymbol{X}$, $\boldsymbol{Y}$ input/output samples

$$\boldsymbol{Y}(k) = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(k) \end{bmatrix} \in \mathbb{R}^k,\ \boldsymbol{X}(k) = \begin{bmatrix} \boldsymbol{x}^\top(1) \\ \boldsymbol{x}^\top(2) \\ \vdots \\ \boldsymbol{x}^\top(k) \end{bmatrix} \in \mathbb{R}^{k\times M}. \quad (3)$$

Sparsity can be induced in the regression problem by a penalty in the norm of $\boldsymbol{w}$, being $\boldsymbol{w}$ the estimate of $\boldsymbol{w}^*$. For example, in ℓ_1-regularization, the cost

$$\min_{\boldsymbol{w}} \left(\boldsymbol{X}(k)\boldsymbol{w} - \boldsymbol{Y}(k)\right)^\top \left(\boldsymbol{X}(k)\boldsymbol{w} - \boldsymbol{Y}(k)\right) + \rho \sum_{i=1}^{M} |\boldsymbol{w}_i|\,, \qquad (4)$$

describes the trade-off between minimizing the error $e = y - \boldsymbol{x}^\top\boldsymbol{w}$, and representing $\boldsymbol{Y}$ as a combination of few non-zero elements in $\boldsymbol{w}$. The tradeoff is regulated by $\rho > 0$.

A. Re-weighting and fused techniques

The literature has shown that the sparsity induced by the ℓ_1-norm can be improved [22]–[24] by introducing positive weights $v_1, v_2, \ldots, v_M$ in the cost (4), that is,

$$\left(\boldsymbol{X}(k)\boldsymbol{w} - \boldsymbol{Y}(k)\right)^\top \left(\boldsymbol{X}(k)\boldsymbol{w} - \boldsymbol{Y}(k)\right) + \rho \sum_{i=1}^{M} v_i\,|w_i|. \qquad (5)$$

The rationale is that the additional weights v_i should make the re-weighted ℓ_1-norm approach the ℓ_0-norm [34]. With

$$v_i = \begin{cases} \frac{1}{|w_i^*|} & \text{if } w_i^* \neq 0 \\ \infty & \text{if } w_i^* = 0, \end{cases} \qquad (6)$$

the re-weighted ℓ_1-norm would coincide with the ℓ_0-norm. To avoid discontinuity and knowledge of $\boldsymbol{w}^*$ in (6), a suitable approximation is obtained via $v_i(k) = 1/(|w_i(k-1)| + \varepsilon)$, with $\boldsymbol{w}^*$ replaced by its latest estimate $w_i(k-1)$, and $\varepsilon > 0$ to allow continuity and avoid division by zero [34].

In fused regression, an extra penalty is included in the cost (4) to measure the correlation between adjacent estimated coefficients [30], [31]. For $w_i(k)$ estimated at iteration k, a possible measure of correlation to be included in (4) is

$$\sum_{i=1}^{M-1} |w_i(k) - w_{i+1}(k)| + \sum_{i=1}^{M-1} |w_i(k) + w_{i+1}(k)|, \qquad (7)$$

which can be suitably approximated as

$$\sum_{i=1}^{M-1} \left| r_i^+(k) w_i(k) + r_i^-(k) w_{i+1}(k) \right|, \qquad (8)$$

with

$$\begin{aligned} r_i^+(k) &= \sigma(w_i(k) + w_{i+1}(k)) + \sigma(w_i(k) - w_{i+1}(k)) \\ r_i^-(k) &= \sigma(w_i(k) + w_{i+1}(k)) - \sigma(w_i(k) - w_{i+1}(k))\,, \end{aligned} \qquad (9)$$

and $\sigma(\cdot)$ is any mirrored sigmoid such as $\sigma(x) = 1/(1+e^{\epsilon x})$, with $\epsilon > 0$ a small constant regulating the transition. Then, the re-weighting technique can be applied once more to the ℓ_1-norm in (8), in a similar fashion as in (5).

III. Proposed Recursive Sparse Regression

When calculating the gradient for minimization of the cost, existing re-weighting algorithms like [23], [24] and zero-attracting algorithms like [28] neglect that the ℓ_1-norm and the re-weighted ℓ_1-norm are non-differentiable. To cope with this issue, we propose a smooth approximation by replacing $|x|$ with $x^2/\sqrt{x^2(k-1) + \varepsilon_d}$, with $\varepsilon_d > 0$. Note that this approximation is consistent with the use of the latest estimate in re-weighting and fused techniques [22]–[24], [30], [31].

A new cost to be minimized is then proposed as

$$\begin{aligned} J &= \left(\boldsymbol{X}(k)\boldsymbol{w} - \boldsymbol{Y}(k)\right)^\top \left(\boldsymbol{X}(k)\boldsymbol{w} - \boldsymbol{Y}(k)\right) \\ &\quad + \rho \sum_{i=1}^{M} \frac{w_i^2}{(|w_i(k-1)| + \varepsilon)\sqrt{w_i^2(k-1) + \varepsilon_d}} \\ &\quad + \gamma \sum_{i=1}^{M-1} \frac{r_i^2(k)}{(|r_i(k-1)| + \varepsilon)\sqrt{r_i^2(k-1) + \varepsilon_d}} \\ &= \left(\boldsymbol{X}(k)\boldsymbol{w} - \boldsymbol{Y}(k)\right)^\top \left(\boldsymbol{X}(k)\boldsymbol{w} - \boldsymbol{Y}(k)\right) \\ &\quad + \rho \boldsymbol{w}^\top \boldsymbol{V}(k) \boldsymbol{w} + \gamma \boldsymbol{w}^\top \boldsymbol{F}^\top(k) \boldsymbol{S}(k) \boldsymbol{F}(k) \boldsymbol{w}, \end{aligned} \qquad (10)$$

where

$$\begin{aligned} \boldsymbol{V}(k) &= \operatorname{diag}\left(v_1(k), \ldots, v_{M-1}(k)\right) \\ v_i(k) &= \frac{1}{(|w_i(k-1)| + \varepsilon)\sqrt{w_i^2(k-1) + \varepsilon_d}} \\ \boldsymbol{S}(k) &= \operatorname{diag}\left(s_1(k), \ldots, s_{M-1}(k)\right) \\ s_i(k) &= \frac{1}{(|r_i(k-1)| + \varepsilon)\sqrt{r_i^2(k-1) + \varepsilon_d}} \\ r_i(k-1) &= r_i^+(k-1) w_i(k-1) + r_i^-(k-1) w_{i+1}(k-1) \\ \boldsymbol{F}(k) &= \begin{bmatrix} r_1^+(k-1) & r_1^-(k-1) & 0 & \cdots & 0 \\ 0 & r_2^+(k-1) & r_2^-(k-1) & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & r_{M-1}^+(k-1) & r_{M-1}^-(k-1) \end{bmatrix}. \end{aligned} \qquad (11)$$

The next step is to minimize the proposed cost to get a sparse estimate of $\boldsymbol{w}^*$. The following result holds.

Theorem 1: The minimization of cost (10) can be performed in a recursive way along the steps in Algorithm 1.

Proof: The partial derivative of (10) wrt $\boldsymbol{w}$ is

$$\nabla J = -\boldsymbol{X}^\top(k)\left(\boldsymbol{Y}(k) - \boldsymbol{X}(k)\boldsymbol{w}\right) + \rho \boldsymbol{V}(k)\boldsymbol{w} + \gamma \boldsymbol{F}^\top(k)\boldsymbol{S}(k)\boldsymbol{F}(k)\boldsymbol{w} = 0, \tag{12}$$

from which we get

$$\boldsymbol{w}(k) = \boldsymbol{P}(k)\boldsymbol{X}^\top(k)\boldsymbol{Y}(k) \tag{13}$$

$$\boldsymbol{P}(k) = \Big(\boldsymbol{X}^\top(k)\boldsymbol{X}(k) + \rho\boldsymbol{V}(k) + \gamma\boldsymbol{F}^\top(k)\boldsymbol{S}(k)\boldsymbol{F}(k)\Big)^{-1}.$$

To obtain a recursive implementation processing only the sample $\boldsymbol{x}(k)$, $y(k)$ instead of the whole input/output batch $\boldsymbol{X}(k)$, $\boldsymbol{Y}(k)$, consider analogously to (13), that

$$\boldsymbol{w}(k-1) = \boldsymbol{P}(k-1)\boldsymbol{X}^\top(k-1)\boldsymbol{Y}(k-1), \tag{14}$$

where a recursive formula for $\boldsymbol{P}^{-1}(k)$ is

$$\begin{aligned}\boldsymbol{P}^{-1}(k) = {} & \boldsymbol{P}^{-1}(k-1) + \boldsymbol{x}(k)\boldsymbol{x}^\top(k) \\ & + \rho\left(\boldsymbol{V}(k) - \boldsymbol{V}(k-1)\right) + \gamma\Big(\boldsymbol{F}^\top(k)\boldsymbol{S}(k)\boldsymbol{F}(k) \\ & -\boldsymbol{F}^\top(k-1)\boldsymbol{S}(k-1)\boldsymbol{F}(k-1)\Big).\end{aligned} \tag{15}$$

Sequential processing arises from manipulating as follows:

$$\begin{aligned}&\boldsymbol{X}^\top(k)\boldsymbol{Y}(k)\\ &= \boldsymbol{X}^\top(k-1)\boldsymbol{Y}(k-1) + \boldsymbol{x}(k)y(k)\\ &= \boldsymbol{P}^{-1}(k-1)\boldsymbol{w}(k-1) + \boldsymbol{x}(k)y(k)\\ &= \left(\boldsymbol{P}^{-1}(k) - \boldsymbol{x}(k)\boldsymbol{x}^\top(k) - \rho\left(\boldsymbol{V}(k) - \boldsymbol{V}(k-1)\right)\right)\boldsymbol{w}(k-1)\\ &\quad - \gamma\Big(\boldsymbol{F}^\top(k)\boldsymbol{S}(k)\boldsymbol{F}(k) - \boldsymbol{F}^\top(k-1)\boldsymbol{S}(k-1)\\ &\quad\quad \boldsymbol{F}(k-1)\Big)\boldsymbol{w}(k-1) + \boldsymbol{x}(k)y(k)\\ &= \boldsymbol{P}^{-1}(k)\boldsymbol{w}(k-1) - \rho\left(\boldsymbol{V}(k) - \boldsymbol{V}(k-1)\right)\boldsymbol{w}(k-1)\\ &\quad - \gamma\Big(\boldsymbol{F}^\top(k)\boldsymbol{S}(k)\boldsymbol{F}(k) - \boldsymbol{F}^\top(k-1)\boldsymbol{S}(k-1)\\ &\quad\quad \boldsymbol{F}(k-1)\Big)\boldsymbol{w}(k-1) + \boldsymbol{x}(k)e(k),\end{aligned} \tag{16}$$

resulting in a recursive formula for $\boldsymbol{w}(k)$

$$\begin{aligned}\boldsymbol{w}(k) = {} & \boldsymbol{w}(k-1) + \boldsymbol{P}(k)\boldsymbol{x}(k)e(k)\\ & - \rho\boldsymbol{P}(k)\left(\boldsymbol{V}(k) - \boldsymbol{V}(k-1)\right)\boldsymbol{w}(k-1)\\ & - \gamma\boldsymbol{P}(k)\Big(\boldsymbol{F}^\top(k)\boldsymbol{S}(k)\boldsymbol{F}(k)\\ & - \boldsymbol{F}^\top(k-1)\boldsymbol{S}(k-1)\boldsymbol{F}(k-1)\Big)\boldsymbol{w}(k-1).\end{aligned} \tag{17}$$

Let $\boldsymbol{G}^{-1}(k) = \boldsymbol{P}^{-1}(k) - \rho\boldsymbol{V}(k) - \gamma\boldsymbol{F}^\top(k)\boldsymbol{S}(k)\boldsymbol{F}(k)$. We obtain a recursive formula for $\boldsymbol{G}^{-1}(k)$ as

$$\boldsymbol{G}^{-1}(k) = \boldsymbol{G}^{-1}(k-1) + \boldsymbol{x}(k)\boldsymbol{x}^\top(k). \tag{18}$$

The matrix inversion lemma[1], applied to (18), allows to obtain a recursive formula for $\boldsymbol{G}(k)$ as

$$\boldsymbol{G}(k) = \boldsymbol{G}(k-1) - \frac{\boldsymbol{G}(k-1)\boldsymbol{x}(k)\boldsymbol{x}^\top(k)\boldsymbol{G}(k-1)}{1 + \boldsymbol{x}^\top(k)\boldsymbol{G}(k-1)\boldsymbol{x}(k)}. \tag{20}$$

Then, using the fact that

$$\boldsymbol{P}^{-1}(k) = \boldsymbol{G}^{-1}(k) + \rho\boldsymbol{V}(k) + \gamma\boldsymbol{F}^\top(k)\boldsymbol{S}(k)\boldsymbol{F}(k), \tag{21}$$

we apply again the matrix inversion lemma to (21) to get a recursive formula for $\boldsymbol{P}(k)$:

$$\boldsymbol{P}(k) = \boldsymbol{U}(k) - \boldsymbol{U}(k)\left(\boldsymbol{U}(k) + \boldsymbol{G}(k)\right)^{-1}\boldsymbol{U}(k), \tag{22}$$

with $\boldsymbol{U}(k) = \Big(\rho\boldsymbol{V}(k) + \gamma\boldsymbol{F}^\top(k)\boldsymbol{S}(k)\boldsymbol{F}(k)\Big)^{-1}$. Thus, all recursions in Algorithm 1 have been derived. ∎

Theorem 1 can be extended in the presence of a forgetting factor $0 < \lambda < 1$, used in re-weighted algorithms like ℓ_1-

[1]The inversion lemma states that for non-singular $\boldsymbol{a} \in \mathbb{R}^{N\times N}$, $\boldsymbol{c} \in \mathbb{R}^{M\times M}$ and $\boldsymbol{b} \in \mathbb{R}^{N\times M}$, $\boldsymbol{d} \in \mathbb{R}^{M\times N}$, the following equality holds

$$(\boldsymbol{a} + \boldsymbol{bcd})^{-1} = \boldsymbol{a}^{-1} - \boldsymbol{a}^{-1}\boldsymbol{b}(\boldsymbol{da}^{-1}\boldsymbol{b} + \boldsymbol{c}^{-1})^{-1}\boldsymbol{da}^{-1} \tag{19}$$

Algorithm 1: Proposed SP-RF-RLS algorithm

Input: Samples $\boldsymbol{x}^\top(k)$, $y(k)$ (collected sequentially)
Init: $\boldsymbol{w}(0) = \boldsymbol{O}_{M\times 1}$, $\boldsymbol{G}(0) = \delta^{-1}\boldsymbol{I}$, $\boldsymbol{V}(0) = \boldsymbol{O}_{M\times M}$
$\boldsymbol{S}(0) = \boldsymbol{O}_{(M-1)\times(M-1)}$, parameters ρ, γ, ε, ε_d, δ
Output: Weight matrix $\boldsymbol{w}$
for *Step* $k = 1; k \leq N$ **do**
  $e(k) = y(k) - \boldsymbol{x}^\top(k)\boldsymbol{w}(k-1)$
  if $k > 1$ **then**
    $\boldsymbol{V}(k) = \operatorname{diag}\left(\frac{\left(|w_1(k-1)| + \varepsilon\right)^{-1}}{\sqrt{w_1^2(k-1) + \varepsilon_d}}, \ldots, \frac{\left(|w_M(k-1)| + \varepsilon\right)^{-1}}{\sqrt{w_M^2(k-1) + \varepsilon_d}}\right)$
    for *Step* $i = 1; i \leq M$ **do**
      $r_i^+(k-1) = \sigma\left(w_i(k-1) + w_{i+1}(k-1)\right) + \sigma\left(w_i(k-1) - w_{i+1}(k-1)\right)$
      $r_i^-(k-1) = \sigma\left(w_i(k-1) + w_{i+1}(k-1)\right) - \sigma\left(w_i(k-1) - w_{i+1}(k-1)\right)$
      $r_i(k-1) = r_i^+(k-1)w_i(k-1) + r_i^-(k-1)w_{i+1}(k-1)$
    end
    $\boldsymbol{S}(k) = \operatorname{diag}\left(\frac{\left(|r_1(k-1)| + \varepsilon\right)^{-1}}{\sqrt{r_1^2(k-1) + \varepsilon_d}}, \ldots, \frac{\left(|r_{M-1}(k-1)| + \varepsilon\right)^{-1}}{\sqrt{r_{M-1}^2(k-1) + \varepsilon_d}}\right)$
    $\boldsymbol{F}(k)$ in (11)
  end
  $\boldsymbol{U}(k) = \Big(\rho\boldsymbol{V}(k) + \gamma\boldsymbol{F}^\top(k)\boldsymbol{S}(k)\boldsymbol{F}(k)\Big)^{-1}$
  $\boldsymbol{G}(k) = \boldsymbol{G}(k-1) - \frac{\boldsymbol{G}(k-1)\boldsymbol{x}(k)\boldsymbol{x}^\top(k)\boldsymbol{G}(k-1)}{1 + \boldsymbol{x}^\top(k)\boldsymbol{G}(k-1)\boldsymbol{x}(k)}$
  $\boldsymbol{P}(k) = \boldsymbol{U}(k) - \boldsymbol{U}(k)\left(\boldsymbol{U}(k) + \boldsymbol{G}(k)\right)^{-1}\boldsymbol{U}(k)$
  $\boldsymbol{w}(k) = \boldsymbol{w}(k-1) + \boldsymbol{P}(k)\boldsymbol{x}(k)e(k) - \rho\boldsymbol{P}(k)\left(\boldsymbol{V}(k) - \boldsymbol{V}(k-1)\right)\boldsymbol{w}(k-1) - \gamma\boldsymbol{P}(k)\Big(\boldsymbol{F}^\top(k)\boldsymbol{S}(k)\boldsymbol{F}(k) - \boldsymbol{F}^\top(k-1)\boldsymbol{S}(k-1)\boldsymbol{F}(k-1)\Big)\boldsymbol{w}(k-1)$
end

RLS and ℓ_1-RRLS [22]–[24]. However, these re-weighted algorithms use a different minimization procedure that makes the ℓ_1 term disappear for $\lambda = 1$ [32], [33]. The minimization procedure in Theorem 1 is such that its ℓ_1 term does *not* disappear for $\lambda = 1$.

It is worth remarking that, due to the approximations involved, SP-RF-RLS cannot be regarded as an exact solution to sparse regression: yet, no extra approximations or assumptions have been adopted other than those in the literature (e.g., re-weighting and fused techniques in Sect. II.A). Yet, we now validate numerically that SP-RF-RLS outperforms state-of-the-art algorithms employing similar approximations and assumptions as those used to develop SP-RF-RLS.

IV. Numerical Validation

We conduct extensive numerical experiments to verify the effectiveness of SP-RF-RLS. The system under consideration has the same form as (2), where the unknown vector $\boldsymbol{w}^*$ has 64 elements. To simulate sparsity, we let only K elements be non-zero, and we set $K = 5, 10, 30, 50$ to test different degrees of sparsity. The locations of the non-zero elements in $\boldsymbol{w}^*$ are random, and their magnitude is random but normalized so that $\|w^*\|_1 = 1$. The input $\boldsymbol{X}$ is taken as 1000 white samples, corrupted by white Gaussian noise $\boldsymbol{n}$. We select different values of signal-to-noise ratio (SNR) between the input $\boldsymbol{X}$ and the observation noise $\boldsymbol{n}$, namely, SNR=$1, 3, 5, 10$dB, to test different degrees of noisy observations. The performance is measured in terms of mean square deviation (MSD):

$$\mathrm{MSD} = E(\|\boldsymbol{w}_{\mathrm{end}} - \boldsymbol{w}^*\|_2^2), \tag{23}$$

where $\boldsymbol{w}_{\mathrm{end}}$ is the estimated $\boldsymbol{w}$ at the end of the 1000 iterations, one for each sample. To obtain an average performance, we perform 100 random trials and average the MSD results.

A. State-of-the-art methods

The state-of-the-art methods used for comparisons are: RLS, ℓ_1-RLS [23], ℓ_1-RRLS [22], ZA-RLS [28], VFF-SMMS [24], SP-R-RLS [33]. To validate the improvements of SP-RF-RLS under the same conditions as proposed in the state of the art, the numerical settings are the same as in [33]. We refer to the literature for the algorithms of these state-of-the-art methods.

Initial conditions and common parameters have been chosen consistently in all algorithms to make the comparisons as fair as possible, e.g., initial estimate $\boldsymbol{w}(0) = 0$, initial covariance $\boldsymbol{P}(0) = \delta^{-1} I_N$ with $\delta = 10^{-3}$, transition constants $\varepsilon = 10^{-1}$, $\varepsilon_d = 10^{-7}$. For SP-RF-RLS, we select $\gamma = \rho$.

B. Analysis of the results

We consider the following experimental scenarios:

1) Different levels of sparsity;
2) Different levels of signal-to-noise ratio;
3) Different levels of regularization;
4) Convergence rate.

TABLE I: Effect of sparsity on MSD(10^{-4})

(SNR=5)	$K=5$	$K=10$	$K=30$	$K=50$
SP-RF-RLS	**3.56**	**3.79**	**4.15**	4.43
SP-R-RLS	3.78	3.94	4.23	**4.40**
RLS	4.53	4.53	4.53	4.53
ℓ_1-RLS	4.52	4.51	4.51	4.51
ℓ_1-RRLS	4.44	4.54	4.58	4.59
ZA-RLS	4.34	4.35	4.47	4.62
VFF-SMMS	4.49	4.49	4.49	4.49
(SNR=10)	$K=5$	$K=10$	$K=30$	$K=50$
SP-RF-RLS	**0.91**	**1.03**	**1.23**	1.39
SP-R-RLS	1.03	1.12	1.28	**1.38**
RLS	1.43	1.43	1.43	1.43
ℓ_1-RLS	1.43	1.43	1.43	1.43
ℓ_1-RRLS	1.37	1.43	1.44	1.45
ZA-RLS	1.34	1.34	1.41	1.52
VFF-SMMS	1.42	1.42	1.42	1.42

TABLE II: Effect of SNR on MSD(10^{-4})

($K=10$)	SNR=1	SNR=3	SNR=5	SNR=10
SP-RF-RLS	**10.21**	**6.25**	**3.79**	**1.03**
SP-R-RLS	10.45	6.43	3.94	1.12
RLS	11.39	7.19	4.53	1.43
ℓ_1-RLS	11.32	7.15	4.51	1.43
ℓ_1-RRLS	11.46	7.22	4.54	1.43
ZA-RLS	11.09	6.95	4.35	1.34
VFF-SMMS	11.28	7.12	4.49	1.42
($K=30$)	SNR=1	SNR=3	SNR=5	SNR=10
SP-RF-RLS	**10.79**	**6.71**	**4.15**	**1.23**
SP-R-RLS	10.92	6.81	4.23	1.28
RLS	11.39	7.19	4.53	1.43
ℓ_1-RLS	11.32	7.14	4.51	1.43
ℓ_1-RRLS	11.51	7.26	4.58	1.44
ZA-RLS	11.27	7.10	4.47	1.41
VFF-SMMS	11.28	7.11	4.49	1.42

1) Different levels of sparsity

While changing the number K of non-zero elements in $\boldsymbol{w}^*$, we consider two values of signal-to-noise ratio (SNR), namely, 5 and 10dB. For a fair comparison, we consider the same regularization parameter $\rho = 1$ for all algorithms, except VFF-SMMS that autonomously updates its regularization parameter. Table I demonstrates that SP-RF-RLS outperforms all the other algorithms, except with the non-sparse scenario $K = 50$ where SP-RF-RLS is the second best, very close to SP-R-RLS.

2) Different levels of signal-to-noise ratio

While changing the values of SNR, we consider two degrees of sparsity, $K = 10$ and $K = 30$. As before, we select a common regularization parameter $\rho = 1$. Table II shows that, although the performance of all algorithms

TABLE III: Effect of regularization on MSD(10^{-4})

(K=10, SNR=5)	$\rho = 0.01$	$\rho = 0.1$	$\rho = 1$	$\rho = 2$
SP-RF-RLS	**4.49**	**4.44**	**3.79**	**3.47**
SP-R-RLS	4.51	4.45	3.94	3.49
ℓ_1-RRLS	4.60	4.59	4.54	4.49
ZA-RLS	4.52	4.50	4.35	4.21
(K=10, SNR=10)	$\rho = 0.01$	$\rho = 0.1$	$\rho = 1$	$\rho = 2$
SP-RF-RLS	**1.35**	**1.31**	**1.03**	**0.88**
SP-R-RLS	1.43	1.39	1.12	0.94
ℓ_1-RRLS	1.45	1.45	1.43	1.40
ZA-RLS	1.43	1.42	1.34	1.26
(K=30, SNR=5)	$\rho = 0.01$	$\rho = 0.1$	$\rho = 1$	$\rho = 2$
SP-RF-RLS	**4.46**	**4.43**	**4.15**	**3.91**
SP-R-RLS	4.55	4.53	4.23	4.00
ℓ_1-RRLS	4.60	4.59	4.58	4.56
ZA-RLS	4.56	4.55	4.47	4.39

naturally decreases as the observations are more and more noisy, the proposed SP-RF-RLS gives the smallest MSD in all scenarios.

3) Different levels of regularization

While changing the regularization parameter ρ, we consider two degrees of sparsity, $K = 10$ and $K = 30$, and two values of signal-to-noise ratio (SNR), 5 and 10dB. We do not report VFF-SMMS because its regularization parameter is updated autonomously, RLS because it has no ℓ_1-regularization, and ℓ_1-RLS because it has similar performance as the reported ℓ_1-RRLS. When changing the regularization parameter, ZA-RLS is the most interesting algorithm for comparison, because changing this parameter changes the zero-attracting effect: a large ρ increases the attraction of the estimate towards zero. The results in Table III show that, although decreasing ρ decreases the performance of all algorithms a bit, the proposed SP-RF-RLS outperforms all methods in all scenarios.

4) Convergence rate

The learning curves for different sparsity and SNR are reported in Figs. 1-2. We only compare SP-R-RLS and the proposed SP-RF-RLS, because it was already shown in [33] that SP-R-RLS converges faster than the state-of-the-art methods used in this study. Figs. 1-2 show that the proposed SP-RF-RLS converges even faster than SP-R-RLS. From Fig. 1, one can notice that the benefits of SP-RF-RLS are greater as the sparsity increases. From Fig. 2, one can notice that the benefits of SP-RF-RLS are greater as the observations are less noisy.

V. Conclusions

This work has presented a recursive implementation of fused sparse regression, with the dataset being processed sequentially rather than as a batch. The proposed algorithm uses a re-weighting technique to deal with the discontinuous nature of ℓ_0-norm, a smooth approximation to deal with the non-differentiable nature of ℓ_1-norm, and a term depending on the difference between the estimated elements to capture structural correlations in the non-zero elements of the sparse model. Comparative experiments have shown that the proposed algorithm outperforms state-of-the-art ones in noisy and sparse scenarios. A performance degradation is noticed only when the scenario is extremely non-sparse.

Interesting future work is to dynamically adjust the regularization parameters according to the estimated level of sparsity, similar to the mechanisms in the VFF-SMMS algorithm [24]. Another interesting direction for future work is to study real-world applications embedding spatial and temporal structure, such as traffic prediction [35].

Acknowledgments

The authors would like to thank mr. Quan Liu for discussions on the state-of-the-art algorithms used in this study.

References

[1] I. Selesnick, "Sparse regularization via convex analysis," *IEEE Transactions on Signal Processing*, vol. 65, no. 17, pp. 4481–4494, 2017.

[2] Y. Wang, J. A. Lopez, and M. Sznaier, "Convex optimization approaches to information structured decentralized control," *IEEE Transactions on Automatic Control*, vol. 63, no. 10, pp. 3393–3403, 2018.

[3] T. Chen, M. S. Andersen, L. Ljung, A. Chiuso, and G. Pillonetto, "System identification via sparse multiple kernel-based regularization using sequential convex optimization techniques," *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 2933–2945, 2014.

[4] C. Novara, "Sparse identification of nonlinear functions and parametric set membership optimality analysis," *IEEE Transactions on Automatic Control*, vol. 57, no. 12, pp. 3236–3241, 2012.

[5] S. Wu, G. Li, L. Deng, L. Liu, D. Wu, Y. Xie, and L. Shi, "L1-norm batch normalization for efficient training of deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 7, pp. 2043–2051, 2018.

[6] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[7] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[8] C. R. Rojas, R. Toth, and H. Hjalmarsson, "Sparse estimation of polynomial and rational dynamical models," *IEEE Transactions on Automatic Control*, vol. 59, no. 11, pp. 2962–2977, 2014.

[9] W. R. Jacobs, T. Baldacchino, T. Dodd, and S. R. Anderson, "Sparse Bayesian nonlinear system identification using variational inference," *IEEE Transactions on Automatic Control*, vol. 63, no. 12, pp. 4172–4187, 2018.

[10] M. Schurch, D. Azzimonti, A. Benavoli, and M. Zaffalon, "Recursive estimation for sparse Gaussian process regression," *Automatica*, vol. 120, p. 109127, 2020.

[11] V. Laurain, R. Toth, D. Piga, and M. A. H. Darwish, "Sparse RKHS estimation via globally convex optimization and its application in LPV-IO identification," *Automatica*, vol. 115, p. 108914, 2020.

[12] G. Pillonetto and A. Yazdani, "Sparse estimation in linear dynamic networks using the stable spline horseshoe prior," *Automatica*, vol. 146, p. 110666, 2022.

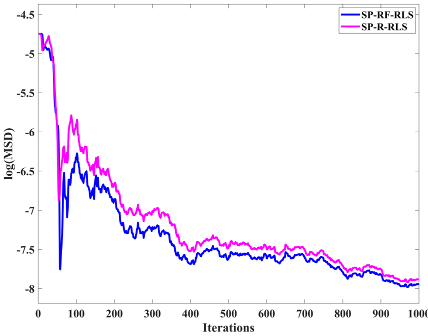
[13] S. O. Haykin, *Adaptive Filter Theory (5th edition)*. Pearson, 2013.

[14] A. H. Sayed, *Fundamentals of adaptive filtering*. John Wiley & Sons, 2003.

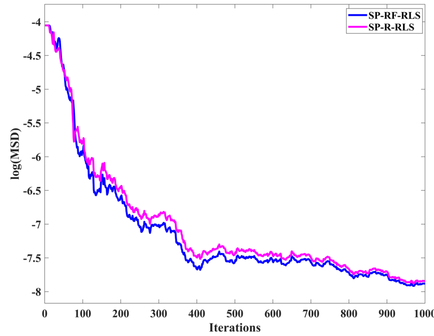
[15] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 508–518, 2000.

[16] Y. Chen, Y. Gu, and A. O. Hero, "Sparse LMS for system identification," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 3125–3128.

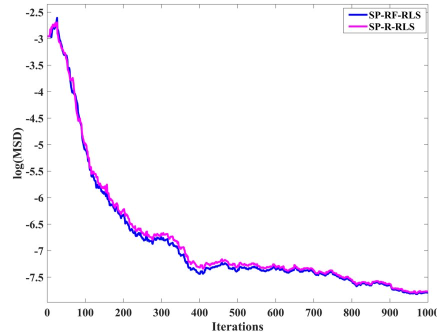
[17] Y. Li and M. Hamamura, "Zero-attracting variable-step-size least mean square algorithms for adaptive sparse channel estimation," *International Journal of Adaptive Control and Signal Processing*, vol. 29, no. 9, pp. 1189–1206, 2015.



(a) $K = 5$

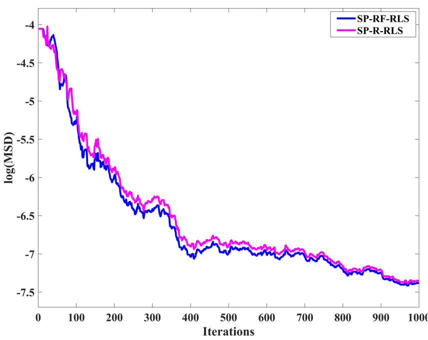


(b) $K = 10$

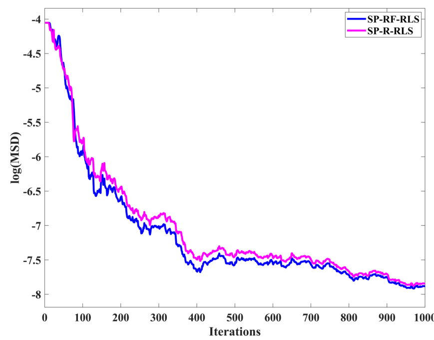


(c) $K = 30$

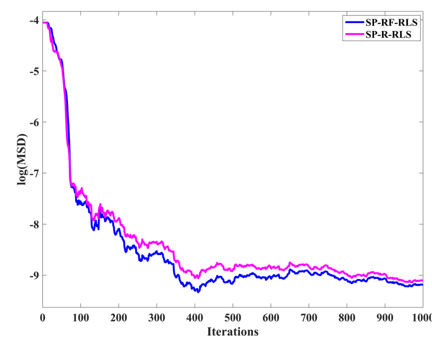
Fig. 1: Learning curves for different levels of sparsity (K is the number of non-zero elements). MSD is in logarithmic scale.



(a) SNR=3



(b) SNR=5



(c) SNR=10

Fig. 2: Learning curves for different levels of signal-to-noise ratio (SNR). MSD is in logarithmic scale.

[18] T. van den Boom, S. Baldi *et al.*, "Online identification of continuous bimodal and trimodal piecewise affine systems," in *2016 European Control Conference (ECC)*. IEEE, 2016, pp. 1075–1070.

[19] K. E. Themelis, A. A. Rontogiannis, and K. D. Koutroumbas, "A variational Bayes framework for sparse adaptive estimation," *IEEE Transactions on Signal Processing*, vol. 62, no. 18, pp. 4723–4736, 2014.

[20] D. Liu, S. Baldi, W. Yu, and C. L. P. Chen, "A hybrid recursive implementation of broad learning with incremental features," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 4, pp. 1650–1662, 2022.

[21] E. M. Ekşioğlu, "RLS adaptive filtering with sparsity regularization," in *10th International Conference on Information Science, Signal Processing and their Applications (ISSPA 2010)*. IEEE, 2010, pp. 550–553.

[22] E. M. Eksioglu, "Sparsity regularised recursive least squares adaptive filtering," *IET Signal Processing*, vol. 5, no. 5, pp. 480–487, 2011.

[23] J. Lim, K. Lee, and S. Lee, "A modified recursive regularization factor calculation for sparse RLS algorithm with l1-norm," *Mathematics*, vol. 9, no. 13, p. 1580, 2021.

[24] Z. F. Li, D. Li, and J. Q. Zhang, "A new penalized recursive least squares method with a variable regularization factor for adaptive sparse filtering," *IEEE Access*, vol. 6, pp. 31 828–31 839, 2018.

[25] B. Dumitrescu, A. Onose, P. Helin, and I. Tabus, "Greedy sparse RLS," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2194–2207, 2012.

[26] E. M. Eksioglu and A. K. Tanc, "RLS algorithm with convex regularization," *IEEE Signal Processing Letters*, vol. 18, no. 8, pp. 470–473, 2011.

[27] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: Where RLS meets the L1-Norm," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3436–3447, 2010.

[28] X. Hong, J. Gao, and S. Chen, "Zero-attracting recursive least squares algorithms," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 1, pp. 213–221, 2016.

[29] P. Aram, V. Kadirkamanathan, and S. R. Anderson, "Spatiotemporal system identification with continuous spatial maps and sparse estimation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 11, pp. 2978–2983, 2015.

[30] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused LASSO," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.

[31] H. Yazdanpanah and J. A. Apolinário Jr, "The extended feature LMS algorithm: exploiting hidden sparsity for systems with unknown spectrum," *Circuits, Systems, and Signal Processing*, vol. 40, no. 1, pp. 174–192, 2021.

[32] D. Liu, S. Baldi, Q. Liu, and W. Yu, "A recursive least squares algorithm with l1 regularization for sparse representation," *Science China Information Sciences*, vol. 66, no. 2, 2023.

[33] Q. Liu, D. Liu, and S. Baldi, "A new recursive approach to sparse representation," in *2023 IEEE International Conference on Development and Learning (ICDL)*, 2023, pp. 461–466.

[34] E. J. Candes, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier and Analysis and Application*, vol. 14, pp. 877–905, 2008.

[35] D. Liu, S. Baldi, W. Yu, J. Cao, and W. Huang, "On training traffic predictors via broad learning structures: A benchmark study," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 2, pp. 749–758, 2022.