Recursive Least-Squares Algorithms for Sparse System Modeling

Hamed Yazdanpanah[1]
Paulo S.R. Diniz[1]

[1]Laboratório de Sinais, Multimídia e Telecomunicações (SMT)
Departamento de Engenharia Eletrônica e de Computação (DEL)
Universidade Federal do Rio de Janeiro (UFRJ)

{hamed.yazdanpanah, diniz}@smt.ufrj.br

International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)

Presentation Outline

1. Introduction
2. Recursive Least-Squares (RLS) Algorithm
3. RLS Algorithm for Sparse Systems (S-RLS)
4. l_0-norm RLS Algorithm for Sparse Systems (l_0-RLS)
5. Data-Selective Version of the Algorithms
6. Results
7. Conclusions

SMT
Sinais, Multimídia e Telecomunicações
COPPE/Poli/UFRJ

UFRJ

Outline

Content

- Two algorithms are proposed in this paper:
  - Recursive Least-Squares algorithm for sparse systems (S-RLS)
  - l_0-norm Recursive Least-Squares algorithm for sparse systems (l_0-RLS)

- Apply data-selective strategy on both algorithms to reduce the computational load

Content

- Two algorithms are proposed in this paper:
  - Recursive Least-Squares algorithm for sparse systems (S-RLS)
  - l_0-norm Recursive Least-Squares algorithm for sparse systems (l_0-RLS)

- Apply data-selective strategy on both algorithms to reduce the computational load

Sparsity & Data-Selective Strategy

Sparsity Modeling/Exploitation

- Usual strategies:
  1. Proportionate update
  2. Sparsity-promoting penalty (l_0 and l_1 norms)
  3. Combination of items 1 and 2
  4. Apply discard function

Data-Selective Strategy

- The output estimation error is small $\Rightarrow$ the current weight vector is acceptable $\Rightarrow$ avoid new update
- Reduce the computational complexity by avoiding unnecessary updates

Sparsity & Data-Selective Strategy

Sparsity Modeling/Exploitation

- Usual strategies:
  1. Proportionate update
  2. Sparsity-promoting penalty (l_0 and l_1 norms)
  3. Combination of items 1 and 2
  4. Apply discard function

Data-Selective Strategy

- The output estimation error is small $\Rightarrow$ the current weight vector is acceptable $\Rightarrow$ avoid new update
- Reduce the computational complexity by avoiding unnecessary updates

Outline

RLS Algorithm: Overview

- Inputs (general case):
  - current data-pair $(\mathbf{x}(k), d(k))$
  - λ forgetting factor
- Problem:

$$\min \xi^d(k) = \sum_{i=0}^{k} \lambda^{k-i}[d(i) - \mathbf{x}^T(i)\mathbf{w}(k)]^2$$

- Solution:

$$\mathbf{w}(k) = \mathbf{S}_D(k)\mathbf{p}_D(k)$$

where

$$\mathbf{S}_D(k) = \frac{1}{\lambda}\left[\mathbf{S}_D(k-1) - \frac{\mathbf{S}_D(k-1)\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{S}_D(k-1)}{\lambda + \mathbf{x}^T(k)\mathbf{S}_D(k-1)\mathbf{x}(k)}\right]$$

$$\mathbf{p}_D(k) = \lambda\mathbf{p}_D(k-1) + d(k)\mathbf{x}(k)$$

Outline

Main Idea

- Existing algorithms for sparse systems include/add something to the classical algorithms $\Rightarrow$ increase complexity

- S-RLS algorithm reduces the importance of coefficients close to zero

- How?
  Applying a discard function

Sinais, Multimídia e Telecomunicações
COPPE/Poli/UFRJ

UFRJ

Main Idea

- Existing algorithms for sparse systems include/add something to the classical algorithms $\Rightarrow$ increase complexity
- S-RLS algorithm reduces the importance of coefficients close to zero
- How?
  Applying a discard function

Main Idea

- Existing algorithms for sparse systems include/add something to the classical algorithms $\Rightarrow$ increase complexity
- S-RLS algorithm reduces the importance of coefficients close to zero
- How?
  Applying a discard function

Discard Function $f_\epsilon(\omega)$

$$f_\epsilon(\omega) \triangleq \begin{cases} \omega & \text{if } |\omega| > \epsilon \\ 0 & \text{if } |\omega| \leq \epsilon \end{cases}$$

ϵ: models the uncertainty on the coefficients

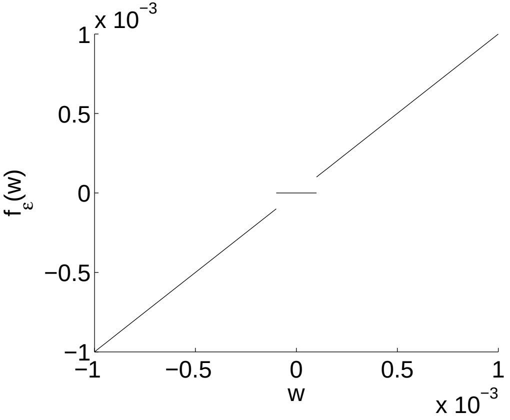


Figure: Discard function $f_\epsilon(\omega)$ for $\epsilon = 10^{-4}$.

Sinais, Multimídia e Telecomunicações
COPPE/Poli/UFRJ

S-RLS Algorithm: Problem and Solution

- Problem:

$$\min \xi^d(k) = \sum_{i=0}^{k} \lambda^{k-i}[d(i) - \mathbf{x}^T(i)\mathbf{f}_\epsilon(\mathbf{w}(k))]^2$$

- Solution:

$$\mathbf{w}(k) = \mathbf{S}_{D,\epsilon}(k)\mathbf{p}_{D,\epsilon}(k)$$

where

$$\mathbf{S}_D(k) = \frac{1}{\lambda}\left[\mathbf{S}_{D,\epsilon}(k-1) - \frac{\mathbf{S}_{D,\epsilon}(k-1)\mathbf{F}_\epsilon(\mathbf{w}(k))\mathbf{x}(k)\mathbf{x}^T(k)\mathbf{F}_\epsilon(\mathbf{w}(k))\mathbf{S}_{D,\epsilon}(k-1)}{\lambda + \mathbf{x}^T(k)\mathbf{F}_\epsilon(\mathbf{w}(k))\mathbf{S}_{D,\epsilon}(k-1)\mathbf{F}_\epsilon(\mathbf{w}(k))\mathbf{x}(k)}\right]$$

$$\mathbf{p}_{D,\epsilon}(k) = \lambda\mathbf{p}_{D,\epsilon}(k-1) + \mathbf{F}_\epsilon(\mathbf{w}(k))\mathbf{x}(k)d(k)$$

$\mathbf{F}_\epsilon(\mathbf{w}(k))$: Jacobian of $\mathbf{f}_\epsilon(\mathbf{w}(k))$

$\Rightarrow$ Diagonal matrix with entries equal to zero or one

$\Rightarrow$ Reduces the importance of small coefficients

Sinais, Multimídia e Telecomunicações
COPPE/Poli/UFRJ

S-RLS Algorithm: Assumptions

Some difficulties that appeared during the derivation of the S-RLS algorithm, and how they were addressed:

- The discard function is not differentiable at the points $\pm\epsilon \Rightarrow$ Solution: use the left and right derivatives
- Matrix $\mathbf{F}_\epsilon(\mathbf{w}(k))$ is not invertible $\Rightarrow$ Solution: replace the zero entries on the diagonal with a small constant
- Initialization cannot be $\mathbf{w}(0) = \mathbf{0}$

Outline

l_0-RLS Algorithm

Problem:

$$\min \xi^d(k) = \sum_{i=0}^{k} \lambda^{k-i}[d(i) - \mathbf{x}^T(i)\mathbf{w}(k)]^2 + \alpha\|\mathbf{w}(k)\|_0$$

Discontinuity of l_0-norm $\Rightarrow$ Geman-McClure function substitutes for l_0-norm

$$\min \xi^d(k) = \sum_{i=0}^{k} \lambda^{k-i}[d(i) - \mathbf{x}^T(i)\mathbf{w}(k)]^2 + \alpha G_\beta(\mathbf{w}(k))$$

Solution:

$$\mathbf{w}(k) = \mathbf{S}_D(k)\left(\mathbf{p}_D(k) - \frac{\alpha}{2}\mathbf{g}_\beta(\mathbf{w}(k-1))\right)$$

where $\mathbf{g}_\beta(\mathbf{w}(k-1)) \triangleq \nabla G_\beta(\mathbf{w}(k-1))$

Sinais, Multimídia e Telecomunicações
COPPE/Poli/UFRJ

Geman-McClure Function $G_\beta(\mathbf{w})$

$G_\beta(\mathbf{w}) \triangleq \sum_{i=0}^{N}\left(1 - \frac{1}{1+\beta|w(i)|}\right)$

β: controls the agreement between the quality of the approximation and smoothness of G_β

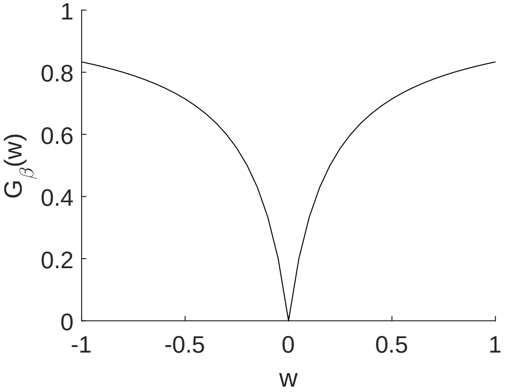


Figure: Geman-McClure function $G_\beta(\mathbf{w})$ for $\beta = 5$.

Outline

Apply Data-Selective Strategy

Data-selective S-RLS (DS-S-RLS) and data-selective l_0-RLS (DS-l_0-RLS) algorithms update whenever the error signal is larger than a predescribed value $\overline{\gamma}$, i.e.,

- DS-S-RLS:

$$\mathbf{w}(k+1) = \begin{cases} \text{implement S} - \text{RLS update} & \text{if } |e(k)| > \overline{\gamma}, \\ \mathbf{w}(k) & \text{otherwise,} \end{cases}$$

- DS-l_0-RLS:

$$\mathbf{w}(k+1) = \begin{cases} \text{implement } l_0 - \text{RLS update} & \text{if } |e(k)| > \overline{\gamma}, \\ \mathbf{w}(k) & \text{otherwise,} \end{cases}$$

Advantage: Avoid unnecessary updates and reduce computational complexity

Sinais, Multimídia e Telecomunicações
COPPE/Poli/UFRJ

Outline

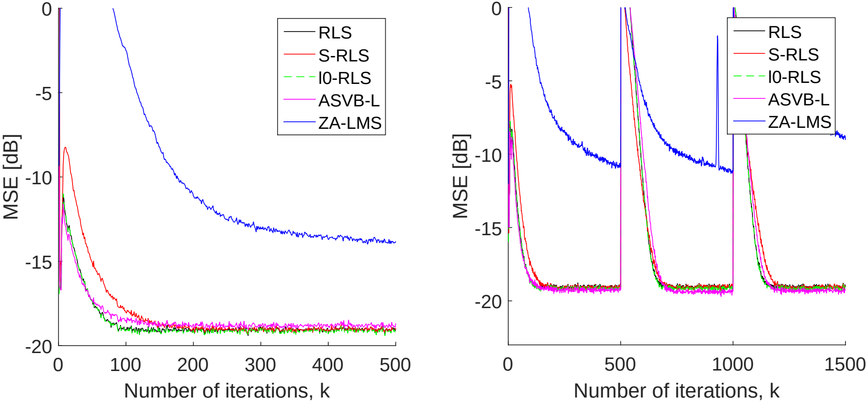
Sinais, Multimídia e Telecomunicações
COPPE/Poli/UFRJ

Scenario: System Identification

- Algorithms tested: RLS, S-RLS, l_0-RLS, Adaptive Sparse Variational Bayes iterative scheme based on Laplace prior (ASVB-L), Zero-Attracting LMS (ZA-LMS), DS-S-RLS, DS-l_0-RLS, DS-ZA-LMS, and DS-ASVB-L algorithms
- Input signal: AR(1) (first-order autoregressive)
- Filter order: $N = 14$
- $\mathbf{w}(0) = [1, \cdots, 1]^T$
- $\epsilon = 0.015 \Rightarrow$ 10 out of 15 coefficients belong to $[-\epsilon, \epsilon]$
- $\beta = 5$
- SNR: 20 dB
- $\overline{\gamma} = \sqrt{5\sigma_n^2}$
- $\lambda = 0.97$

Algorithms without Data-Selective Strategy

- Learning (MSE) curves



(a) (b)

Figure: (a) Time-invariant sparse system; (b) time-variant sparse system.

SMT
Sinais, Multimídia e Telecomunicações
COPPE/Poli/UFRJ

UFRJ

Algorithms with Data-Selective Strategy

- Learning (MSE) curves
- Update rate: DS-ZA-LMS: 44.5%, DS-S-RLS: 10.3%, DS-l_0-RLS: 9.8%, DS-ASVB-L: 7.9%.

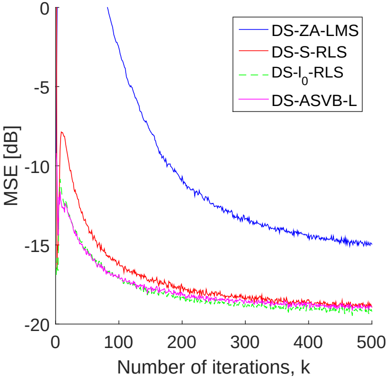


Figure: Learning curves of DS algorithms for time-invariant sparse system.

Sinais, Multimídia e Telecomunicações
COPPE/Poli/UFRJ

Outline

Conclusions

- In this presentation:
  - The concept of discard function is used in order to propose S-RLS algorithm
  - The l_0-norm and Geman-McClure function are used in order to propose l_0-RLS algorithm
  - The proposed algorithms have better performance compared to the LMS-based algorithms (e.g., ZA-LMS)
  - The proposed algorithms demand lower computational load compared to the Bayesian algorithms (e.g., ASVB-L)
  - Data-selective strategy reduces extremely the computational complexity

Thank You!