RLS ADAPTIVE FILTERING WITH SPARSITY REGULARIZATION

Ender M. Ekşioğlu

Istanbul Technical University
Department of Electronics and Communications Engineering
Istanbul, Turkey

ABSTRACT

We propose a new algorithm for the adaptive identification of sparse systems. The algorithm is based on the minimization of the RLS cost function when regularized by adding a sparsity inducing ℓ_1 norm penalty. The resulting recursive update equations for the system impulse response estimate are in a similar form to the regular RLS. However, they include novel terms which account for the sparsity prior. The proposed, ℓ_1 relaxation based RLS algorithm emphasizes sparsity during the adaptive filtering process and allows for faster convergence when the system under consideration is sparse. Computer simulations comparing the performance of the proposed algorithm to conventional RLS and other adaptive algorithms are provided. Simulations demonstrate that the new algorithm exploits the inherent sparse structure effectively.

Index Terms— Adaptive filters, RLS, sparsity.

1. INTRODUCTION

Sparse adaptive filtering, where the impulse response for the system to be identified is assumed to be of a sparse form has acquired attention recently. The sparsity prior has applications in acoustic and network echo cancellation [1] and communication channel identification [2]. Numerous adaptive algorithms building upon the a priori knowledge of the sparsity of the system to be identified have been developed. Partial-update adaptive algorithms and proportionate adaptive algorithms [3] are two well-known approaches to the problem. In these algorithms, least mean square (LMS) type adaptive algorithms have been modified as to incorporate the sparsity prior. Partial-update and proportionate adaptive algorithms offer performance improvement when compared to plain LMS algorithms by utilizing the additional knowledge of sparsity. Recently, novel LMS type algorithms which incorporate the sparsity condition directly into the cost function have been developed [4–6]. The common idea in these papers is to add a penalty term in the form of an ℓ_p norm of the weight vector, into the overall cost function to be minimized. In [4], the additional penalty is $\|\mathbf{w}(n)\|_p,\ 0 < p \leqslant 1$. Here $\|(\cdot)\|_p$ denotes the ℓ_p norm and $\mathbf{w}(n)$ is the instantaneous weight vector. Minimizing the cost function results in a novel steepest descent algorithm and corresponding LMS type adaptive algorithms. [5] pursuits a similar idea, but here the penalties considered are in the forms of ℓ_1 norm or a log-sum term. The ℓ_1 norm penalty results in an adaptive algorithm called as the zero-attracting LMS (ZA-LMS) [5]. In [6], the penalty in the cost function is defined by an approximation to the ℓ_0 norm, which is a count for the total number of nonzero terms. By replacing $\|\mathbf{w}(n)\|_0$ with an analytic approximation, the minimization problem for the cost function becomes tractable. The end result is an algorithm denoted as ℓ_0-LMS. As can be seen from the references, sparsity based adaptive algorithms have been mostly confined to the LMS domain.

Recursive least squares (RLS) adaptive filtering is another important modality in the adaptive system identification setting, which has in general an order of magnitude faster convergence rate than LMS algorithm. However, RLS has been scarcely applied in sparse system identification. Attempts at an RLS algorithm for sparse system identification and sparse signal estimation are given in [7] and [8], respectively. In both papers, the RLS cost function is modified by the addition of a weighted version of the convex ℓ_1 norm $\|\mathbf{w}(n)_1\|$. In [7], the optimal solution minimizing this cost function is found by an online and adaptive version of the expectation maximization (EM) algorithm, which is called as the Low Complexity Recursive $\mathcal{L}_1-$Regularized Least Squares (SPARLS) algorithm. The SPARLS algorithm is developed by modifying the EM algorithm for the minimization of the convex cost function to the case of the streaming data. In [8], the same modified cost function is utilized for sparse signal estimation problem. Subgradient-based iterative minimization is utilized for the estimation of the possibly time varying sparse signal. In both papers, the update procedure for the weight (or signal) vector is not in a similar form to the regular RLS case, where the update is in the well-known simple form $\mathbf{w}(n) = \mathbf{w}(n-1) + \Delta[\mathbf{w}(n-1)]$.

In this paper, we propose an RLS adaptive algorithm for sparse system identification. The algorithm will utilize the modified RLS cost function with an additional sparsity inducing ℓ_1 penalty term. Here, we rather start from scratch and find the recursive minimization procedure in a manner similar to the conventional approach, namely finding the gradient and setting it equal to zero. We find a subgradient vector of the cost function and try to make it zero. At the end we come up with an algorithm very much similar to the regular RLS algorithm. The only difference is in the weight vector update equation, where a novel zero-attracting, sparsity inducing additional term is included. We will call this new algorithm as the ℓ_1-RLS.

Firstly, we give a brief outline of the sparse adaptive system identification setting. Then, we develop the novel ℓ_1-RLS algorithm by outlining the similarities to the development of regular RLS. We give the final form of ℓ_1-RLS algorithm. Next, we present simulation results comparing the novel ℓ_1-RLS algorithm to regular RLS, regular LMS and ZA-LMS algorithms. The steady-state error and tracking performance of these algorithms are compared via learning curves, where the system to be identified is assumed to have a sparse and time-invariant impulse response.

2. ℓ_1-RLS ALGORITHM

Let us consider the system identification setting given by the following input-output equation.

$$y(n) = \mathbf{h}^T \mathbf{x}(n) + \eta(n) \tag{1}$$

Here, $\mathbf{h} = [h_0,\ h_1, \ldots, h_{N-1}]^T$ is the sparse system tab weight vector, $\mathbf{x}(n) = [x(n), x(n-1), \ldots x(n-N+1)]^T$ is the input signal vector and $\eta(n)$ denotes the observation noise. The aim of the adaptive system identification algorithm is to estimate the system parameters $\mathbf{h}$ from the input and output signals in a sequential manner. We denote the tab weight estimate at time n as $\mathbf{h}(n)$. In conventional RLS, the cost function to be minimized by the weight estimate is given by

$$\mathcal{E}(n) = \sum_{m=0}^{n} \lambda^{n-m} |e(m)|^2. \tag{2}$$

λ is the exponential weighting constant and $e(n)$ is the instantaneous error term.

$$e(n) = y(n) - \mathbf{h}^T(n)\mathbf{x}(n) \tag{3}$$

In this work, we assume that the underlying filter coefficient vector $\mathbf{h}$ has a sparse form . Hence, we want to modify the cost function in a manner that underlines this a priori information. A tractable way to force sparsity is by using the ℓ_1-norm of the weight vector. Hence, we regularize the RLS cost function by including the weighted ℓ_1 norm of the current tab estimate as a sparsifying term.

$$J(n) = \frac{1}{2}\mathcal{E}(n) + \gamma \|\mathbf{h}(n)\|_1 \tag{4}$$

Here, $\gamma > 0$ is a parameter that governs the tradeoff between sparsity and estimation error. $\|\mathbf{h}(n)\|_1$ is the ℓ_1 norm of the weight vector and is given by

$$\|\mathbf{h}(n)\|_1 = \sum_{k=0}^{N-1} |h_k(n)| \tag{5}$$

We want to minimize this regularized cost function $J(n)$ with respect to the filter tab weights. Let $\hat{\mathbf{h}}(n)$ denote the optimal least squares estimate for the tab weight vector which minimizes $J(n)$. In the standard RLS case when the cost function is simply $\mathcal{E}(n)$, the minimization condition is written in terms of the gradient of $\mathcal{E}(n)$ with respect to $\mathbf{h}(n)$.

$$\nabla \mathcal{E}(n) \mid_{\hat{\mathbf{h}}(n)} = 2 \frac{\partial \mathcal{E}(n)}{\partial \mathbf{h}^*(n)}\Big|_{\hat{\mathbf{h}}(n)} = \mathbf{0} \tag{6}$$

However, the ℓ_1 norm term $\|\mathbf{h}(n)\|_1$ in (5) and hence $J(n)$ in (4) are nondifferentiable at any point where $h_k(n) = 0$. A substitute for the gradient in the case of nondifferentiable convex functions such as $\|\mathbf{h}(n)\|_1$ here is offered by the definition of the subgradient [9, p. 227]. The set of all the subgradients of some convex function $f(x)$ is called as the subdifferential. The subdifferential is denoted by $\partial f(x)$. The subdifferential for $\|\mathbf{h}(n)\|_1$ is calculated as follows.

$$\partial \|\mathbf{h}(n)\|_1 = \Big\{ \mathbf{d} \mid \|\mathbf{d}\|_\infty \leqslant 1, \mathbf{d}\, \mathbf{h}(n) = \|\mathbf{h}(n)\|_1 \Big\} \tag{7}$$

Hence, the k^{th} element of the subdifferential for $\|\mathbf{h}(n)\|_1$ can be written in the below form.

$$\Big\{\partial \|\mathbf{h}(n)\|_1\Big\}_k = \begin{cases} \big\{h_k/|h_k|\big\}, & h_k \neq 0 \\ \big\{d \mid |d| \leqslant 1\big\}, & h_k = 0 \end{cases} \tag{8}$$

For any point with $h_k = 0$, there is a valid subgradient vector with its k^{th} entry equal to zero. Using these results we can state that one valid subgradient vector for $\|\mathbf{h}(n)\|_1$ is as given below.

$$\nabla^S \|\mathbf{h}(n)\|_1 = \operatorname{sgn}\big(\mathbf{h}(n)\big) \tag{9}$$

∇^S denotes a subgradient at the corresponding point. $\operatorname{sgn}(\cdot)$, acting possibly on a vector, denotes the componentwise sign function. One subgradient vector of the penalized cost function $J(n)$ in (4) with respect to the weight vector $\mathbf{h}(n)$ can be written using (9) and the fact that $\mathcal{E}(n)$ is differentiable everywhere.

$$\nabla^S J(n) = \frac{1}{2}\nabla \mathcal{E} + \gamma \operatorname{sgn}\big(\mathbf{h}(n)\big) \tag{10}$$

The i^{th} element of this vector is calculated as below [10].

$$\Big\{\nabla^S J(n)\Big\}_i = -\sum_{m=0}^{n} \lambda^{n-m} e(m) x^*(m-i+1) + \gamma \operatorname{sgn}\big(h_i(n)\big) \tag{11}$$

We set the subgradient equal to zero to find the optimal least squares solution, namely $\hat{\mathbf{h}}(n)$.

$$-\sum_{m=0}^{n} \lambda^{n-m} \Big\{ y(m) - \sum_{k=0}^{N-1} \hat{h}_k(n) x(m-k) \Big\} x^*(m-i+1) = -\gamma \operatorname{sgn}\big(\hat{h}_i(n)\big) \tag{12}$$

The above equation, after some manipulation, assumes the form below.

$$\sum_{k=0}^{N-1} \hat{h}_k(n) \Big\{ \sum_{m=0}^{n} \lambda^{n-m} x(m-k) x^*(m-i+1) \Big\} = \sum_{m=0}^{n} \lambda^{n-m} y(m) x^*(m-i+1) - \gamma \operatorname{sgn}\big(\hat{h}_i(n)\big) \tag{13}$$

(13), written for all $i = 1, \dots, N$ together in a matrix form, results in the modified deterministic normal equations.

$$\mathbf{\Phi}(n)\hat{\mathbf{h}}(n) = \mathbf{r}(n) - \gamma\,\mathrm{sgn}\big(\hat{\mathbf{h}}(n)\big) \tag{14}$$

Here, $\mathbf{\Phi}(n)$ is the exponentially weighted deterministic autocorrelation matrix estimate. $\mathbf{r}(n)$ is the deterministic cross-correlation estimate between $y(n)$ and $\mathbf{x}(n)$. These two quantities can be updated by rank-one recursive equations.

$$\mathbf{\Phi}(n) = \lambda\mathbf{\Phi}(n-1) + \mathbf{x}^*(n)\mathbf{x}^T(n) \tag{15}$$

$$\mathbf{r}(n) = \lambda\mathbf{r}(n-1) + y(n)\mathbf{x}^*(n) \tag{16}$$

With a slight change in notation the normal equation (14) becomes

$$\mathbf{\Phi}(n)\hat{\mathbf{h}}(n) = \boldsymbol{\theta}(n) \tag{17}$$

where $\boldsymbol{\theta}(n) = \mathbf{r}(n) - \gamma\,\mathrm{sgn}\big(\hat{\mathbf{h}}(n)\big)$. The $\boldsymbol{\theta}(n)$ term can also be described by a recursive equation, namely

$$\begin{aligned}\boldsymbol{\theta}(n) = \lambda\boldsymbol{\theta}(n-1) + y(n)\mathbf{x}^*(n) - \\ \Big\{\gamma\,\mathrm{sgn}\big(\hat{\mathbf{h}}(n)\big) - \lambda\gamma\,\mathrm{sgn}\big(\hat{\mathbf{h}}(n-1)\big)\Big\}.\end{aligned} \tag{18}$$

In a similar vein to the conventional RLS paradigm, instead of solving the normal equations for the optimal least squares solution $\hat{\mathbf{h}}(n)$ directly, we search for an iterative solution of the form

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mathbf{\Delta}\hat{\mathbf{h}}(n-1). \tag{19}$$

Here, $\mathbf{\Delta}\hat{\mathbf{h}}(n-1)$ is an instantaneous corrective step. To reach such a solution, we have to modify (18) into a recursion with only $\hat{\mathbf{h}}(n-1)$ terms on the right side. To this end, we assume that the sign of the weight values do not change significantly in a single time step. Hence, we approximate (18) by

$$\boldsymbol{\theta}(n) \approx \lambda\boldsymbol{\theta}(n-1) + y(n)\mathbf{x}^*(n) + \gamma(\lambda-1)\,\mathrm{sgn}\big(\hat{\mathbf{h}}(n-1)\big) \tag{20}$$

The normal equation (17) can be rewritten as

$$\hat{\mathbf{h}}(n) = \mathbf{P}(n)\boldsymbol{\theta}(n) \tag{21}$$

where $\mathbf{P}(n)$ is the inverse of the autocorrelation matrix.

$$\mathbf{P}(n) = \mathbf{\Phi}^{-1}(n)$$

We insert the recursions (15) and (20) into (21) to come up with the following result.

$$\begin{aligned}\hat{\mathbf{h}}(n) = \mathbf{P}(n-1)\boldsymbol{\theta}(n-1) - \mathbf{k}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)\boldsymbol{\theta}(n-1) \\ + y(n)\mathbf{k}(n) + \gamma\Big(\frac{\lambda-1}{\lambda}\Big)\times \\ \Big\{\mathbf{P}(n-1)\,\mathrm{sgn}\big(\hat{\mathbf{h}}(n-1)\big) - \mathbf{k}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)\,\mathrm{sgn}\big(\hat{\mathbf{h}}(n-1)\big)\Big\}\end{aligned}$$

Here, $\mathbf{k}(n)$ is the gain vector.

$$\mathbf{k}(n) = \frac{\mathbf{P}(n-1)\mathbf{x}^*(n)}{\lambda + \mathbf{x}^H(n)\mathbf{P}(n-1)\mathbf{x}(n)} \tag{22}$$

Algorithm 1 ℓ_1 regularized RLS (ℓ_1-RLS) algorithm.

$\lambda,\ \gamma,\ x(n),\ y(n)$ ▷ inputs

$\mathbf{h}(-1) = \mathbf{0}, \quad \mathbf{P}(-1) = \delta^{-1}\mathbf{I}$ ▷ initial values

for $n := 0, 1, 2, \dots$ **do** ▷ time recursion

$\quad \mathbf{k}_\lambda(n) = \mathbf{P}(n-1)\mathbf{x}^*(n)$

$\quad \mathbf{k}(n) = \dfrac{\mathbf{k}_\lambda(n)}{\lambda + \mathbf{x}^T(n)\mathbf{k}_\lambda(n)}$

$\quad \xi(n) = y(n) - \mathbf{h}^T(n-1)\mathbf{x}(n)$

$\quad \mathbf{P}(n) = \dfrac{1}{\lambda}\Big[\mathbf{P}(n-1) - \mathbf{k}(n)\mathbf{k}_\lambda^H(n)\Big]$

$\quad \mathbf{h}(n) = \mathbf{h}(n-1) + \mathbf{k}(n)\xi(n) + \gamma\Big(\dfrac{\lambda-1}{\lambda}\Big)\Big\{\mathbf{I}_N - \mathbf{k}(n)\mathbf{x}^T(n)\Big\}\mathbf{P}(n-1)\mathrm{sgn}\big(\mathbf{h}(n-1)\big)$

end for ▷ end of recursion

Using the matrix inversion lemma, it can be shown that the time update for the inverse correlation matrix can be performed by the well known Riccati equation.

$$\mathbf{P}(n) = \lambda^{-1}\Big\{\mathbf{P}(n-1) - \mathbf{k}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)\Big\} \tag{23}$$

By realizing that $\hat{\mathbf{h}}(n-1) = \mathbf{P}(n-1)\boldsymbol{\theta}(n-1)$, the recursive update for the tab weight vector assumes its final form.

$$\begin{aligned}\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mathbf{k}(n)\Big\{y(n) - \hat{\mathbf{h}}^T(n-1)\mathbf{x}(n)\Big\} + \\ \gamma\Big(\frac{\lambda-1}{\lambda}\Big)\Big\{\mathbf{I}_N - \mathbf{k}(n)\mathbf{x}^T(n)\Big\}\mathbf{P}(n-1)\mathrm{sgn}\big(\hat{\mathbf{h}}(n-1)\big)\end{aligned} \tag{24}$$

This update equation finalizes the ℓ_1-RLS algorithm. The overall algorithm is given in Alg. 1. When we compare the ℓ_1-RLS weight update with the regular RLS update equation, we see that the last term starting with $\gamma\big(\frac{\lambda-1}{\lambda}\big)$ constitutes the difference from regular RLS. If we choose $\lambda = 1$ or $\gamma = 1$, for this formulation the ℓ_1-RLS coincides with regular RLS.

3. SIMULATION RESULTS

In this section we compare the performance of the novel ℓ_1-RLS algorithm to the regular RLS, regular LMS and one other sparsity oriented adaptive algorithm. The first experiment considers the tracking capabilities of ℓ_1-RLS, RLS, ZA-LMS [5] and LMS algorithms under white excitation. The sparse system to be identified has a total of 64 tabs and 4 of them are nonzero. The positions and amplitudes of the nonzero tab weights are chosen randomly. AWGN observation noise resulting in an SNR of 20 dB is added to the system output. The four algorithms are realized for a total of 500 runs and the mean square deviation (MSD) of the system impulse response estimate versus time iteration index is plotted. The parameters for the different algorithm are chosen as follows.

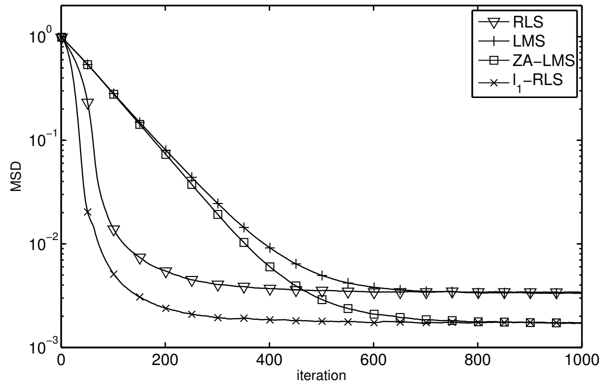

Fig. 1. Learning curves for ℓ_1-RLS, RLS, ZA-LMS and LMS.

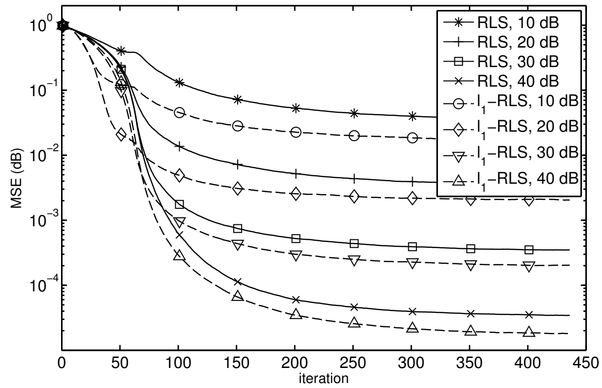

Fig. 2. Learning curves for ℓ_1-RLS and RLS for SNR=40, 30, 20 and 10 dB.

- ℓ_1-RLS and RLS: $\lambda = 0.99$; ℓ_1-RLS: $\gamma = 3$
- ZA-LMS and LMS: $\mu = 0.008$
- ZA-LMS: $\rho = 3 \times 10^{-4}$, $\sigma = 10$

The λ and μ parameters are chosen as to result in approximately equal steady-state MSD's for RLS and LMS. The γ and ρ parameters are found by repeated trials as to produce the minimum steady-state MSD for their respective algorithms. The results are given in Fig.1. ℓ_1-RLS presents convergence and steady-state error improvements over the regular RLS algorithm, just as ZA-LMS works better than the regular LMS algorithm.

In the second experiment we compare the performance of the novel ℓ_1-RLS algorithm to the regular RLS under different SNR values. The sparse system is constructed as in the first experiment. The learning curves for 40, 30, 20 and 10 dB SNR are presented in Fig.2. The corresponding γ values for ℓ_1-RLS are 0.3, 0.5, 3 and 5, respectively. $\lambda = 0.99$ for both ℓ_1-RLS and regular RLS are utilized throughout the simulations. The ℓ_1-RLS has better convergence and steady-state properties than the regular RLS.

4. CONCLUSIONS

This paper introduced a new RLS algorithm, namely ℓ_1-RLS, applicable for the adaptive identification of systems with sparse impulse response. The novel update equations for this algorithm are developed by regularizing the cost function with an ℓ_1 norm term. Numerical simulations demonstrate that the algorithm indeed brings about better convergence and steady state performance than regular RLS when the system to be identified is sparse. Topics for future work might include theoretical analysis for the steady state error and simulations studying performance of the proposed algorithm in the case of sparse, slowly time-varying systems.

5. REFERENCES

[1] P. Loganathan, A. W. H. Khong, and P. A. Naylor, "A class of sparseness-controlled algorithms for echo cancellation," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 17, no. 8, pp. 1591–1601, Nov. 2009.

[2] C. Carbonelli, S. Vedantam, and U. Mitra, "Sparse channel estimation with zero tap detection," *IEEE Trans. Wireless Commun.*, vol. 6, no. 5, pp. 1743–1763, May 2007.

[3] K. Doğançay and P. A. Naylor, "Recent advances in partial update and sparse adaptive filters," in *Proc. European Signal Processing Conference EUSIPCO*, 2005.

[4] B. D. Rao and B. Song, "Adaptive filtering algorithms for promoting sparsity," in *Proc. ICASSP*, vol. 6, 6–10 April 2003, pp. VI361–VI364.

[5] Y. Chen, Y. Gu, and A. O. Hero, "Sparse LMS for system identification," in *Proc. ICASSP*, 19–24 April 2009, pp. 3125–3128.

[6] Y. Gu, J. Jin, and S. Mei, "l_0 norm constraint LMS algorithm for sparse system identification," *IEEE Signal Processing Letters*, vol. 16, no. 9, pp. 774–777, Sept. 2009.

[7] B. Babadi, N. Kalouptsidis, and V. Tarokh, "Comparison of SPARLS and RLS algorithms for adaptive filtering," in *Proc. IEEE Sarnoff Symposium, SARNOFF'09*, March 30 2009–April 1 2009, pp. 1–5.

[8] D. Angelosante and G. B. Giannakis, "RLS-weighted Lasso for adaptive estimation of sparse signals," in *Proc. ICASSP*, 19–24 April 2009, pp. 3245–3248.

[9] D. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex Analysis and Optimization*. Athena Scientific, Cambridge, Massachusetts, 2003.

[10] S. Haykin, *Adaptive Filter Theory (4th Ed.)*. Prentice Hall, 2001.