Combination of recursive supervised and semisupervised filters for improved unbiased estimation

Jerónimo Arenas-García #1, Carlos Moriana-Varo #2, Jan Larsen *3

Dep. Signal Theory and Communications, Universidad Carlos III de Madrid
28911 Leganés-Madrid, Spain
1 jarenas@tsc.uc3m.es
2 cmoriana@tsc.uc3m.es

* Dep. Informatics and Mathematical Modelling, The Technical University of Denmark
2800 Kgs. Lyngby, Denmark
3 jl@imm.dtu.dk

***Abstract*—In this paper we investigate the steady-state performance of semisupervised regression models adjusted using a modified RLS-like algorithm, identifying the situations where the new algorithm is expected to outperform standard RLS. By using an adaptive combination of the supervised and semisupervised methods, the resulting adaptive filter is guaranteed to perform at least as well as the best contributing filter, therefore achieving universal performance. The analysis and behavior of the methods is illustrated through a set of examples in a plant identification setup, analyzing both steady-state and convergence situations.**

I. Introduction

Adaptive filters (AFs) have become very popular tools that offer a convenient solution (mainly) to estimation problems, and are widely used in many signal processing applications. Although they can be applied in stationary situations, AFs become especially convenient for time-varying scenarios, since their iterative nature allows them to track changing solutions [1], [2].

Among the existing techniques, in this paper we will devote our attention to the Recursive Least Squares (RLS) algorithm, which can be obtained as the solution to a well-defined least squares problem. Introduction of a forgetting factor and an exponential window allow the RLS filter to forget the past, and to adapt its parameters to the newly arriving data. An efficient implementation of the filter can be obtained by exploiting the Matrix Inversion Lemma (also known as the Sherman-Morrison-Woodbury formula).

The solution provided by the RLS filter is composed of two terms, one of them depending on the input data only, while the second implies both input and target data. In this paper, we study a modified RLS-like algorithm where the input signal auto-correlation matrix is estimated without applying any forgetting factor. We refer to this algorithm as semi-supervised RLS (RSS), in the sense that unlabeled data (or even *a priori* information about the input autocorrelation matrix) can be incorporated to the adjustment of the filter weights. Stationary performance of the RSS is studied in steady-state, identifying the conditions under which this filter is expected to outperform standard RLS.

Combination of adaptive filters have recently gained attention in the Adaptive Filter literature as an alternative to varying step size strategies, and as a valid approach to improve AF performance in general. Different combination algorithms and corresponding analyses can be found in [3]–[7]. In this paper, we used the combination scheme of [3] as a way to prevent the degradation that can be incurred by the RSS filter with respect to RLS under certain conditions. Since the combination offers steady-state universal performance, this scheme can be expected to perform as well as the best component filter. Additionally, we will find out that the combination can simultaneously outperform both component filters in certain situations.

The performance of both the basic RSS and the RLS-RSS combination is studied theoretically, and illustrated through several examples in a system identification setup.

II. Semisupervised RLS algorithm

The RLS filter offers a closed form exact solution of the following cost function:

$$J[\mathbf{w}(n)] = \sum_{i=1}^{n} \lambda^{n-i}[d(i) - y(n,i)]^2 \tag{1}$$

where λ is a forgetting factor, $\mathbf{w}(n)$ are the filter weights at time n, $d(n)$ is the desired signal, and $y(n,i) = \mathbf{w}^T(n)\boldsymbol{u}(i)$ is the output of the filter at time n when processing the input regressor received at the ith iteration, $\boldsymbol{u}(i)$.

The solution provided by the RLS filter is in the form of the product of the inverse of an estimation of the autocorrelation matrix of the input data, $\boldsymbol{R} = \mathbb{E}\{\boldsymbol{u}(n)\boldsymbol{u}^T(n)\}$, and of the cross-correlation between the input and desired data, $\boldsymbol{z} = \mathbb{E}\{d(n)\boldsymbol{u}(n)\}$. To be more precise,

$$\mathbf{w}_{\text{RLS}}(n) = \boldsymbol{P}(n)\hat{\boldsymbol{z}}(n), \tag{2}$$

where

$$\boldsymbol{P}(n) = \hat{\boldsymbol{R}}^{-1}(n) = \left[\sum_{i=1}^{n} \lambda^{n-i}\boldsymbol{u}(i)\boldsymbol{u}^T(i)\right]^{-1} \quad (3)$$

$$\hat{\boldsymbol{z}}(n) = \sum_{i=1}^{n} \lambda^{n-i}\boldsymbol{u}(i)d(i) \quad (4)$$

Practical implementations of RLS avoid the inversion of $\hat{\boldsymbol{R}}(n)$ at every iteration by applying the Matrix Inversion Lemma to (3), so that matrix $\boldsymbol{P}(n)$ is recursively updated instead. In this way, matrix inversion is replaced by a division by a scalar. Note also that $\hat{\boldsymbol{R}}(n)$ as given by (3) is singular at early iterations. To avoid this problem, $\boldsymbol{P}(0)$ is typically initialized to the identity matrix multiplied by a large constant.

As we have just seen, the standard RLS algorithm assumes a unique forgetting factor for the estimations of $\boldsymbol{R}$ and $\boldsymbol{z}$. In this paper, we decouple the time constants of both estimations, and replace (3) and (4) by

$$\boldsymbol{P}_{\text{RSS}}(n) = \left(\frac{1-\lambda_P^n}{1-\lambda_P}\right)\left[\sum_{i=1}^{n} \lambda_P^{n-i}\boldsymbol{u}(i)\boldsymbol{u}^T(i)\right]^{-1} \quad (5)$$

$$\boldsymbol{z}_{\text{RSS}}(n) = \left(\frac{1-\lambda^n}{1-\lambda}\right)\sum_{i=1}^{n} \lambda^{n-i}\boldsymbol{u}(i)d(i) \quad (6)$$

These changes make sense when the input signal statistics, and the cross-correlation between the input and desired output change at different speeds. One of the most evident examples of this situation, and the one we will consider most in this paper, occurs when the input signal is stationary but the optimal weight solution is time-varying, thus making $\boldsymbol{z}(n) = \mathbb{E}\{d(n)\boldsymbol{u}(n)\}$ also a function of time. Such situation is likely to occur in system identification setups, where the system can change, but the input signal statistics remain unaffected. When this is the case, one could opt to keep the forgetting factor in (6) smaller than one, while using a larger value for λ_P. In the limit, when $\lambda_P \to 1$, (5) becomes

$$\boldsymbol{P}_{\text{RSS}}(n) = \left[\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{u}(i)\boldsymbol{u}^T(i)\right]^{-1} \quad (7)$$

For an efficient implementation, $\boldsymbol{P}_{\text{RSS}}(n)$ can still be updated at each iteration using rank 1 updates and the Matrix Inversion formula.

Using (7) and (6), the newly proposed semisupervised RLS-like algorithm computes the filter weights as

$$\mathbf{w}_{\text{RSS}}(n) = \boldsymbol{P}_{\text{RSS}}(n)\boldsymbol{z}_{\text{RSS}}(n), \quad (8)$$

We refer to this algorithm as a semisupervised algorithm, in the sense that estimation of $\boldsymbol{P}_{\text{RSS}}(n)$ can be carried out using both labeled and unlabeled data, whereas $\boldsymbol{z}_{\text{RSS}}$ requires labeled patterns. In fact, (7) converges to the inverse of the true autocorrelation matrix as $n \to \infty$. If such information were *a priori* available, it could be directly incorporated into the solution.

In the following section, we present an exact analysis about RSS steady-state performance.

III. RSS PERFORMANCE ANALYSIS

A. Data model and definitions

In the sequel we adopt the following assumptions:

a) $d(n)$ and $\boldsymbol{u}(n)$ are related via a linear regression model
$$d(n) = \mathbf{w}_o^T\boldsymbol{u}(n) + e_0(n),$$
where $\mathbf{w}_o$ is a constant weight vector of length M relating the input and output signals, and σ_0^2 is the power of the zero-mean Gaussian i.i.d. noise $e_0(n)$.

b) Additive noise $e_0(n)$ is independent of $\boldsymbol{u}(n), \forall n$.

c) $\boldsymbol{u}(n)$ are i.i.d. multivariate Gaussian, i.e., $\boldsymbol{u}(n) \sim N(\boldsymbol{0}, \boldsymbol{R})$. It follows from this and the previous assumption that $d(n) \sim N(0, \sigma_0^2 + \mathbf{w}_o^T\boldsymbol{R}\mathbf{w}_o)$.

To measure filter performance, it is customary to use the excess mean-square-error (EMSE), which is defined as the excess over the minimum mean-square error that can be achieved by a filter of length M, namely σ_0^2. It can easily be shown (see, e.g., [1]) that the EMSE of the filter can be expressed as

$$\text{EMSE}(n) = \mathbb{E}\{(\mathbf{w}_o - \mathbf{w}_{\text{RSS}}(n))^T\boldsymbol{u}(n)\boldsymbol{u}^T(n)(\mathbf{w}_o - \mathbf{w}_{\text{RSS}}(n))\}.$$

Since in the case of i.i.d. regressors $\boldsymbol{u}(n)$ is independent of $\mathbf{w}_{\text{RSS}}(n)$, the EMSE can alternatively be expressed as

$$\text{EMSE}(n) = \text{Tr}\left[\boldsymbol{R}\mathbb{E}\{(\mathbf{w}_o - \mathbf{w}_{\text{RSS}}(n))(\mathbf{w}_o - \mathbf{w}_{\text{RSS}}(n))^T\}\right]. \quad (9)$$

B. Steady-state analysis

We start by noting that, as $n \to \infty$, (7) converges to the inverse of $\boldsymbol{R}$ with probability one. Therefore, for sufficiently large n, RSS solution can be replaced by

$$\mathbf{w}_{\text{RSS}}(n) = \boldsymbol{R}^{-1}\boldsymbol{z}_{\text{RSS}}(n) \quad (10)$$

Introducing this expression and (6) into the expectation term in (9) leads to

$$\begin{aligned}\mathbb{E}\{(\mathbf{w}_o - \mathbf{w}_{\text{RSS}}(n))(\mathbf{w}_o - \mathbf{w}_{\text{RSS}}(n))^T\} &= \\ -\mathbf{w}_o\mathbf{w}_o^T + \mathbb{E}\{\mathbf{w}_{\text{RSS}}(n)\mathbf{w}_{\text{RSS}}^T(n)\} &= \\ -\mathbf{w}_o\mathbf{w}_o^T + \boldsymbol{R}^{-1}\mathbb{E}\{\boldsymbol{z}_{\text{RSS}}(n)\boldsymbol{z}_{\text{RSS}}^T(n)\}\boldsymbol{R}^{-1} & \quad (11)\end{aligned}$$

After some algebra, and using well-known expressions for the second- and fourth-order moments of Gaussian variables, the expression above simplifies to

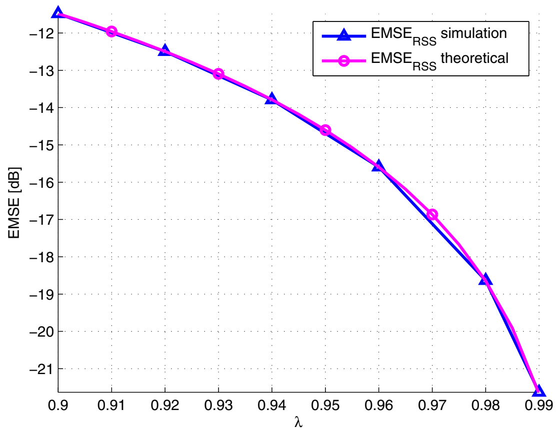
$$\begin{aligned}\mathbb{E}\{(\mathbf{w}_o - \mathbf{w}_{\text{RSS}}(n))(\mathbf{w}_o - \mathbf{w}_{\text{RSS}}(n))^T\} &= \\ N^{-1}(\lambda, n)[\mathbf{w}_o\mathbf{w}_o^T + (\sigma_0^2 + \mathbf{w}_o^T\boldsymbol{R}\mathbf{w}_o)\boldsymbol{R}^{-1}] & \quad (12)\end{aligned}$$

where

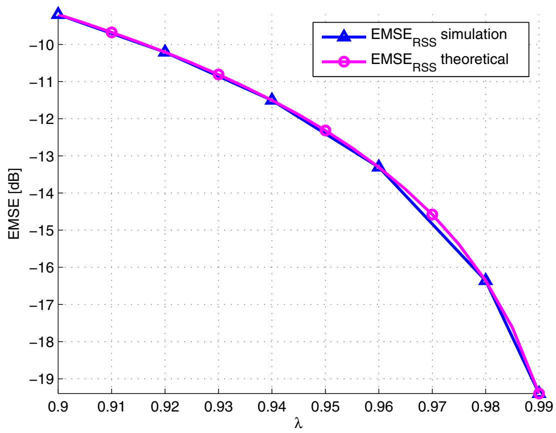
$$N(\lambda, n) = \frac{1+\lambda}{1-\lambda}\frac{1-\lambda^n}{1+\lambda^n}.$$

Finally, taking the limit of (9) as $n \to \infty$, and noting that $N(\lambda) = \lim_{n\to\infty} N(\lambda, n) = \frac{1+\lambda}{1-\lambda}$, the steady-state EMSE of the RSS algorithm is given by

$$\text{EMSE}_{\text{RSS}}(\infty) = \frac{1-\lambda}{1+\lambda}\left[M\sigma_0^2 + (M+1)\mathbf{w}_o^T\boldsymbol{R}\mathbf{w}_o\right]. \quad (13)$$



(a)



(b)

Fig. 1. Theoretical and simulated steady-state EMSE of the RSS algorithm as a function of the forgetting factor λ: (a) $M = 32$; (b) $M = 128$.

With the goal of showing the validity of this expression, we have carried out experiments for a wide range of forgetting factors, estimating the steady-state EMSE of the RSS algorithm by averaging over 30000 iterations after filter convergence, and over 100 independent realizations. Figure 1 represents simulated and theoretical EMSE of the RSS algorithm for $\boldsymbol{R} = \mathbf{I}$, $\sigma_0^2 = 0.01$, and for two different filter lengths, $M = 32$ and $M = 128$. As it can be seen, (13) provides an appropriate model for RSS steady-state performance for a wide range of λ and M.

IV. Combination of RLS and RSS algorithms

When compared to the EMSE of an RLS filter, the RSS algorithm is found to be superior only for large M and/or small forgetting factors λ. Thus, in many practical situations RSS will actually degrade the performance with respect to the application of standard RLS. In order to avoid this very undesired behavior, one could opt to combine RLS and RSS filters with a common forgetting factor λ. By doing so, we could obtain a superior performance from RSS whenever possible, while using the standard RLS in the opposite situation.

In this paper we use the convex combination scheme of [3], that obtains the output of the overall filter as

$$y(n) = \eta(n) y_{\text{RLS}}(n) + [1 - \eta(n)] y_{\text{RSS}}(n) \tag{14}$$

where $\eta(n)$ is a mixing parameter that controls the combination. In order to keep $\eta(n)$ between 0 and 1, it is defined as the output of a *sigmoid* function,

$$\eta(n) = \frac{1}{1 + e^{-a(n)}}.$$

At each iteration, $a(n)$ is adapted to minimize the square error of the overall filter, and then $\eta(n)$ is recovered from the current value of $a(n)$. Please, refer to [3] for further details on this combination scheme.

In [3] it was shown that the considered combination scheme is universal in steady-state, i.e., the combination performs at least as the best component filter. Furthermore, it was shown that under some conditions the combination of two filters can simultaneously outperform both component filters. This behavior can be explained by a low correlation between the errors of the filters, which results in a reduction of the error variance by averaging the outputs of the component filters. This can be the case for the proposed combination of RLS and RSS, as we will see in the experiments section.

V. Experiments

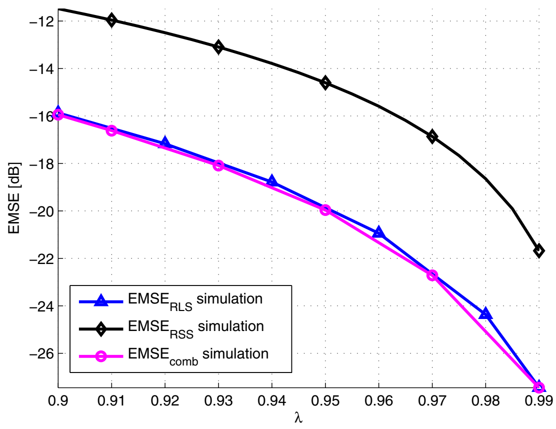
In this section, we will illustrate the performance of both the RSS algorithm and the RLS-RSS combination in a plant identification setup. For the experiments, the input regressors $\boldsymbol{u}(n)$ are i.i.d. colored Gaussian vectors, with zero mean, and variance adjusted so that $\text{Tr}\{\boldsymbol{R}\} = M$. The output additive noise $e_0(n)$ is i.i.d. zero mean Gaussian, with variance $\sigma_0^2 = 0.01$. The optimal solution $\mathbf{w}_o$ is generated randomly, and its norm adjusted so that at the output of the filter we get the following signal-to-noise ratio:

$$\text{SNR} = \frac{\mathbf{w}_o^T \boldsymbol{R} \mathbf{w}_o}{\sigma_0^2} = \frac{1}{M\sigma_0^2}$$

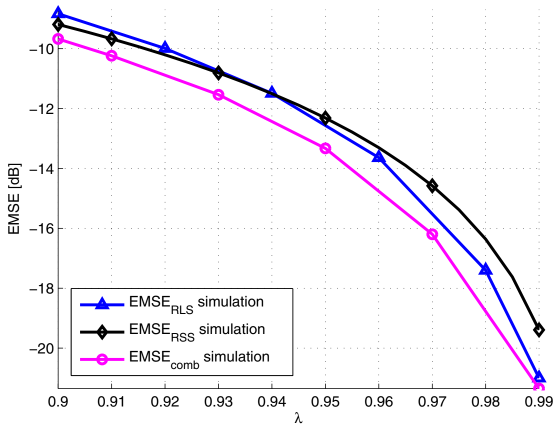
.

Figure 2 shows the steady-state EMSE of all filters for different values of M and λ. For small M, the RLS filter systematically outperforms RSS, but the RLS-RSS combination remains robust in this situation, performing exactly like the RLS filter in this case. For $M = 128$, however, the performances of both component filters become very similar, with RSS achieving a slightly smaller EMSE than RLS for $\lambda < 0.94$. With respect to the combination, we see that it is able to simultaneously outperform both component filters for the whole explored range of λ.

Figure 3 shows the EMSE time evolution for all studied algorithms. After the initial convergence, the optimal solution $\mathbf{w}_o$ is abruptly changed at $n = 500$ and $n = 2500$ to study the ability of the algorithms to reconverge. The norm of $\mathbf{w}_o$ is adjusted to get an initial $\text{SNR} = 1/(\sqrt{4M}\sigma_0^2)$, which is then changed to $\text{SNR} = 1/(4M\sigma_0^2)$ and $\text{SNR} = 4/(M\sigma_0^2)$ after each of the changes.

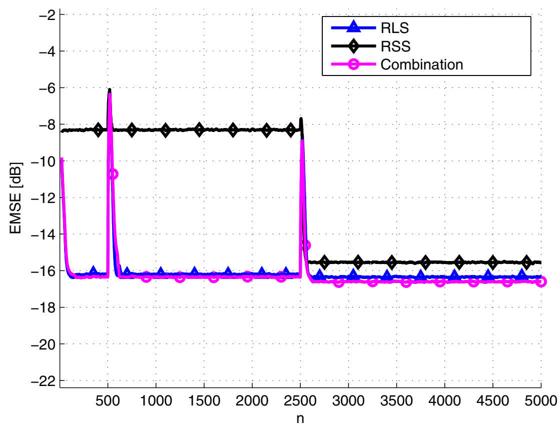


(a)

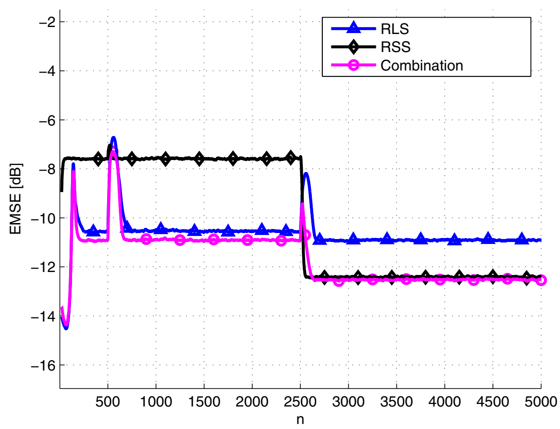


(b)

Fig. 2. Steady-state EMSE of the RLS and RSS algorithms, as well as for their combination, as a function of the forgetting factor λ: (a) $M = 32$; (b) $M = 128$.



(a)



(b)

Fig. 3. EMSE evolution for the RLS and RSS algorihtms, as well as for their combination. After the initial convergence, the optimal solution $\mathbf{w}_o$ changes at $n = 500$ and $n = 2500$. (a) $M = 32$ and $\lambda = 0.96$. (b) $M = 128$ and $\lambda = 0.96$.

As we can see in Subfigure 3(a), RLS outperforms RSS during the whole experiment. Nevertheless, the combination remains robust to this fact, and it follows the RLS component in this situation. Furthermore, after $n = 2500$ we can see that the combination slightly outperforms the RLS component, thus showing a better-than-universal behavior. As we have already stated, this effect can be explained by the ability of the combination to exploit the low cross-correlation between the component filters to reduce the variance of the estimated output.

When the optimum solution length is increased to $M = 128$ (Subfigure 3(b)), RSS performance becomes closer to that of the RLS algorithm, especially for low SNR (i.e., after $n = 2500$). RLS-RSS combination in this case is able to switch between RLS and RSS as necessary, so that the combination outperforms both component filters when considering the whole experiment. Again, a better-than-universal behavior is observed during $n < 2500$.

VI. Conclusions

This paper presents a new algorithm for semisupervised adaptive filtering. The new algorithm is similar to standard RLS, but decouples the forgetting factors for the estimations of the inverse of $\boldsymbol{R}$ and for the cross-correlation $\boldsymbol{z}(n)$. This is especially convenient in situations where the optimum solution changes over time, but the input signal statistics remain unchanged. In such situations, one can keep a long (even infinite) window for a better estimation of the input signal autocorrelation, while using a faster adaptation for the cross-correlation term.

Steady-state performance of the RSS algorithm has been studied both theoretically and through several simulation examples. In order to avoid the possible performance degradation of RSS with respect to RLS, a combination of RLS and RSS can be used. It has been demonstrated empirically that the resulting combination scheme performs like the best component, and possibly better than any of them when certain conditions are met.

Future work will include the study of these schemes when tracking time-varying $\mathbf{w}_o$, and compare the performance of the proposed algorithms with a combination of two RLS filters with different forgetting factors.

Acknowledgment

The authors acknowledge support from the IST Programme of the European Community under the PASCAL2 Network of Excellence IST-2007-216886. The work of Arenas-García was also partly supported by MEC project TEC2008-02473.

References

[1] A. H. Sayed, *Fundamentals of Adaptive Filtering*, Wiley-Interscience, 2003.

[2] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 2002.

[3] J. Arenas-García, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1078–1090, Mar. 2006.

[4] N. J. Bershad, J. C. M. Bermudez, and J.-Y. Tourneret, "An affine combination of two LMS adaptive filters—transient mean-square analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1853–1864, May 2008.

[5] M. T. M. Silva and V. H. Nascimento, "Improving the tracking capability of adaptive filters via convex combination," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3137–3149, July 2008.

[6] R. Candido, M. T. M. Silva, and V. H. Nascimento, "Affine combinations of adaptive filters," in *Conf. Rec. of the 42nd Asilomar Conf. on Sign., Syst. & Comp.*, 2008.

[7] L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, and J. Arenas-García, "A normalized adaptation scheme for the convex combination of two adaptive filters," in *Proc., ICASSP 2008*, Las Vegas, NV, USA, pp. 3301–3304.