ONLINE SYSTEM IDENTIFICATION UNDER NON-NEGATIVITY AND ℓ_1-NORM CONSTRAINTS ALGORITHM AND WEIGHT BEHAVIOR ANALYSIS

Jie Chen [†‡], *Cédric Richard* [†], *Henri Lantéri* [†], *Céline Theys* [†], *Paul Honeine* [‡]

[†]Université de Nice Sophia-Antipolis, CNRS, Observatoire de la Côte d'Azur, Nice, France
{cedric.richard, henri.lateri, celine.theys}@unice.fr
[‡] Institut Charles Delaunay, Université de Technologie de Troyes, CNRS, STMR, Troyes, France
{jie.chen, paul.honeine}@utt.fr

ABSTRACT

Information processing with ℓ_1-norm constraint has been a topic of considerable interest during the last five years since it produces sparse solutions. Non-negativity constraints are also desired properties that can usually be imposed due to inherent physical characteristics of real-life phenomena. In this paper, we investigate an online method for system identification subject to these two families of constraints. Our approach differs from existing techniques such as projected-gradient algorithms in that it does not require any extra projection onto the feasible region. The mean weight-error behavior is analyzed analytically. Experimental results show the advantage of our approach over some existing algorithms. Finally, an application to hyperspectral data processing is considered.

1. INTRODUCTION

Information processing with ℓ_1-norm constraint has been a topic of considerable interest during the last five years. Many methods in statistical learning, compressive sensing and related fields have been formulated as convex optimization problems subject to ℓ_1-norm constraint on the set of parameters to estimate. One motivation for using ℓ_1-norm constraint is that it produces sparse solutions [1]. Non-negativity constraints are also desired properties that can usually be imposed due to inherent physical characteristics of real-life phenomena. For instance, in the study of a concentration field or a thermal radiation field, non-negativity of observations is a property that can be advantageously exploited by sensor networks [2]. Combining non-negativity and ℓ_1-norm constraints has been addressed within various contexts including, e.g., image deblurring in astrophysics [3] and hyperspectral data analysis [4].

In this paper, we consider the problem of minimizing a convex cost function subject to non-negativity constraint and constant ℓ_1-norm constraint, called hereafter full-additivity constraint, within the context of online system identification. Popular approaches are the projected-gradient type methods and the regularization-based methods. The former project the results onto the feasible region at each iteration [5, 6]. The latter minimize the objective function with additional penalty terms that are related to the constraints [7]. This paper is an extension of our previous work [2, 8] in the sense that, in addition to the non-negativity constraint, we also consider now the full-additivity constraint. We propose a strategy that differs from projected-gradient methods and regularization methods in the sense that it integrates the constraints into the coefficient update process. No extra projection procedure or penalty term is required, and constraints are strictly satisfied at each iteration. In order to deal with non-stationary systems, and to reduce computational requirements, we also develop a stochastic gradient algorithm that updates the parameters in an online way. Its mean-weight convergence behavior is analyzed analytically. Simulations show the advantage of the proposed algorithm over some existing methods, and the accuracy of the analysis.

2. PROBLEM FORMULATION

Consider an unknown system, characterized by a set of real valued discrete-time responses to known stationary inputs. In this paper, we address the problem of deriving a filter $\hat{y} = \boldsymbol{x}^\top(n)\,\boldsymbol{\alpha}(n)$, subject to constraints (2)-(3), that best models the desired output $y(n)$ corrupted by noise $z(n)$. See Figure 1. Input signal $x(n)$ is zero-mean, and $z(n)$ is a zero-mean additive noise uncorrelated with any other signal. The system identification problem can be formulated as

$$\boldsymbol{\alpha}^o = \arg\min_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}) \tag{1}$$

$$\text{subject to} \quad \alpha_i \geq 0, \qquad i = 1, \ldots, L \tag{2}$$

$$\sum_{i=1}^{L} \alpha_i = c_0 \tag{3}$$

with $J(\alpha)$ a convex cost function, L the order of the filter, and c_0 a given positive constant.

3. WEIGHT UPDATE ALGORITHM

3.1 Learning with non-negativity constraints

In [2, 8], we have studied an algorithm for system identification involving only the constraint (2), without the full-additivity constraint (3). For this problem, using Lagrange multiplier method, the Karush-Kuhn-Tucker conditions at the optimum were combined into the following expression [2, 3]

$$\alpha_i^o \left[-\nabla_\alpha J(\boldsymbol{\alpha}^o)\right]_i = 0 \tag{4}$$

where the minus sign is just used to make a gradient descent of the criterion $J(\boldsymbol{\alpha})$ apparent. Equations of

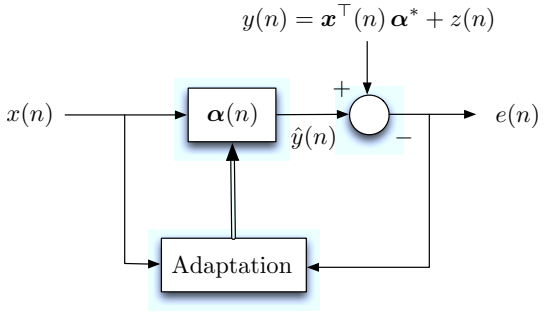

Figure 1: Adaptive system identification under constraints

the form $g(u) = 0$ can be solved with a fixed-point algorithm, under some conditions on function g, by considering the problem $u = u + g(u)$. Implementing this fixed-point strategy with equation (4) leads us to the component-wise gradient descent algorithm

$$\alpha_i(k+1) = \alpha_i(k) + \mu_i(k)\,\alpha_i(k)[-\nabla_\alpha J(\boldsymbol{\alpha}(k))]_i \quad (5)$$

where $\mu_i(k) > 0$ is a positive step-size that allows us to control convergence. Suppose that $\alpha_i(k) > 0$. The non-negativity of $\alpha_i(k+1)$ is guaranteed if, and only if,

$$1 + \mu_i(k)\,[-\nabla_\alpha J(\boldsymbol{\alpha}(k))]_i > 0. \quad (6)$$

If $[\nabla_\alpha J(\boldsymbol{\alpha}(k))]_i < 0$, condition (6) is satisfied and the non-negativity constraint does not impose any restriction on the step-size. Conversely, if $[\nabla_\alpha J(\boldsymbol{\alpha}(k))]_i > 0$, non-negativity of $\alpha_i(k+1)$ holds if

$$0 < \mu_i(k) \leq \frac{1}{[\nabla_\alpha J(\boldsymbol{\alpha})]_i}. \quad (7)$$

In [8], we have shown that condition (6) is not necessary for convergence of the algorithm (5) to the solution of problem (1)-(2).

3.2 Algorithm for fully constrained problem

If non-negativity of $\boldsymbol{\alpha}(n)$ is guaranteed at each iteration, we can make the following variable change to ensure that the full-additivity constraint (3) is satisfied

$$\alpha_j = \frac{w_j}{\sum_{\ell=1}^L w_\ell}\, c_0. \quad (8)$$

With $w_i \geq 0$, the problem becomes unconstrained with respect to constraint (3). The partial derivative of the cost function J with respect to the new variables w_i can be expressed as follows

$$\frac{\partial J}{\partial w_i} = \sum_{j=1}^L \frac{\partial J}{\partial \alpha_j} \times \frac{\partial \alpha_j}{\partial w_i} \quad (9)$$

with

$$\frac{\partial \alpha_j}{\partial w_i} = \frac{c_0\delta_{ij} - \alpha_j}{\sum_{\ell=1}^L w_\ell} \quad (10)$$

and δ_{ij} the Kronecker symbol. Replacing (10) into (9), the partial derivative of J with respect to w_i can now be written as

$$\frac{\partial J}{\partial w_i} = \frac{-1}{\sum_{\ell=1}^L w_\ell}\left(-c_0\frac{\partial J}{\partial \alpha_i} + \sum_{j=1}^L \alpha_j\left(\frac{\partial J}{\partial \alpha_j}\right)\right). \quad (11)$$

Let us now use the same rule as (5) for updating the non-negative entries $w_i(k)$. The component-wise update equation is given by

$$w_i(k+1) = w_i(k) + \ldots$$
$$+\,\mu\,\frac{w_i(k)}{\sum_{\ell=1}^L w_\ell(k)}\left(-c_0\frac{\partial J}{\partial \alpha_i(k)} + \sum_{j=1}^L \alpha_j(k)\left(\frac{\partial J}{\partial \alpha_j(k)}\right)\right)$$

It can be verified that $\sum_{i=1}^L w_i(k+1) = \sum_{i=1}^L w_i(k)$ for all μ by summing both sides from $i = 1$ to L. This yields

$$w_i(k+1) = w_i(k) + \ldots$$
$$+\,\mu\, w_i(k)\left(-c_0\frac{\partial J}{\partial \alpha_i(k)} + \sum_{j=1}^L \alpha_j(k)\left(\frac{\partial J}{\partial \alpha_j(k)}\right)\right).$$

where $1/\sum_{\ell=1}^L w_\ell(k)$ has been absorbed into μ since it is constant across iterations.

Dividing by $\sum_{i=1}^L w_i(k+1)$ and $\sum_{i=1}^L w_i(k)$ the left and right hand sides of the above equation, respectively, and considering the variable change defined by (8), we finally obtain

$$\alpha_i(k+1) = \alpha_i(k) + \ldots$$
$$+\,\mu\alpha_i(k)\left(-c_0\frac{\partial J}{\partial \alpha_i(k)} + \sum_{j=1}^L \alpha_j(k)\left(\frac{\partial J}{\partial \alpha_j(k)}\right)\right). \quad (12)$$

We can verify that $\sum_{i=1}^L \alpha_i(k+1) = \sum_{i=1}^L \alpha_i(k)$, which means that the algorithm satisfies the full-additivity constraint as long as the weight vector is initialized by any non-negative vector $\boldsymbol{\alpha}(0)$ whose sum of its components equals c_0.

3.3 Application to mean square error criterion

A general update rule was derived in (12) for any convex cost function $J(\boldsymbol{\alpha})$. We shall now consider the usual situation where the mean square error is considered, namely

$$J(\boldsymbol{\alpha}) = E[(\boldsymbol{\alpha}^\top\,\boldsymbol{x}(n) - y(n))^2]. \quad (13)$$

Using criterion (13) in (12) requires that the correlation matrix $\boldsymbol{R}_x$ and the intercorrelation vector $\boldsymbol{r}_{xy}$ be estimated. In order to enable online computation, we shall now present a LMS-style algorithm based on the stochastic gradient

$$-\nabla J(\boldsymbol{\alpha}) \approx \boldsymbol{x}(n)\,e(n)$$

with $e(n) = y(n) - \boldsymbol{x}^\top(n)\,\boldsymbol{\alpha}(n)$. Substituting this expression into equation (12) yields the following update rule, written in vectorial form,

$$\boldsymbol{\alpha}(n+1) = \boldsymbol{\alpha}(n) + \ldots$$
$$+\,\mu\,\mathrm{diag}\{\boldsymbol{\alpha}(n)\}\left(c_0\boldsymbol{x}(n)e(n) - \mathbf{1}\,\boldsymbol{\alpha}^\top(n)\boldsymbol{x}(n)e(n)\right) \quad (14)$$

where $\mathbf{1}$ is the all-one vector, and $\mathrm{diag}\{\cdot\}$ is a diagonal matrix.

4. MEAN WEIGHT CONVERGENCE BEHAVIOR

We shall now propose a model to characterize the stochastic behavior of the online algorithm (14). As it is rather challenging to study the performance of such iterative systems, several simplifying assumptions will be made. However, experiments will show that the simulated results and the model outputs match almost perfectly. We define the weight-error vector

$$\boldsymbol{v}(n) = \boldsymbol{\alpha}(n) - \boldsymbol{\alpha}^* \tag{15}$$

where $\boldsymbol{\alpha}^*$ is the solution of the unconstrained problem. See Figure 1. Replacing $\boldsymbol{\alpha}(n)$ by $\boldsymbol{v}(n) + \boldsymbol{\alpha}^*$ in (14), and writing $e(n) = z(n) - \boldsymbol{v}^\top(n)\,\boldsymbol{x}(n)$, leads us to the following expression

$$\begin{aligned}
\boldsymbol{v}(n+1) &= \boldsymbol{v}(n) + \ldots \\
&+\mu\Big(\underbrace{c_0\, z(n)\, \mathrm{diag}\{\boldsymbol{x}(n)\}\, \boldsymbol{v}(n)}_{\boldsymbol{u}_1} + \underbrace{c_0\, z(n)\, \mathrm{diag}\{\boldsymbol{x}(n)\}\, \boldsymbol{\alpha}^*}_{\boldsymbol{u}_2} \\
&- \underbrace{c_0\, \mathrm{diag}\{\boldsymbol{x}(n)\}\, \boldsymbol{v}(n)\, \boldsymbol{v}^\top(n)\, \boldsymbol{x}(n)}_{\boldsymbol{u}_3} \\
&- \underbrace{c_0\, \mathrm{diag}\{\boldsymbol{x}(n)\}\, \boldsymbol{\alpha}^*\, \boldsymbol{x}^\top(n)\, \boldsymbol{v}(n)}_{\boldsymbol{u}_4} \\
&- \underbrace{\boldsymbol{\alpha}^*\, \boldsymbol{\alpha}^{*\top} \boldsymbol{x}(n)\, z(n)}_{\boldsymbol{u}_5} - \underbrace{\boldsymbol{\alpha}^*\, \boldsymbol{v}^\top(n)\, \boldsymbol{x}(n)\, z(n)}_{\boldsymbol{u}_6} \\
&- \underbrace{\boldsymbol{v}(n)\, \boldsymbol{\alpha}^{*\top}\, \boldsymbol{x}(n)\, z(n)}_{\boldsymbol{u}_7} - \underbrace{\boldsymbol{v}(n)\, \boldsymbol{v}^\top(n)\, \boldsymbol{x}(n)\, z(n)}_{\boldsymbol{u}_8} \\
&+ \underbrace{\boldsymbol{v}^\top(n)\, \boldsymbol{x}(n)\, \boldsymbol{\alpha}^*\, \boldsymbol{\alpha}^{*\top}\, \boldsymbol{x}(n)}_{\boldsymbol{u}_9} + \underbrace{\boldsymbol{v}^\top(n)\, \boldsymbol{x}(n)\, \boldsymbol{\alpha}^*\, \boldsymbol{v}^\top(n)\, \boldsymbol{x}(n)}_{\boldsymbol{u}_{10}} \\
&+ \underbrace{\boldsymbol{v}^\top(n)\, \boldsymbol{x}(n)\, \boldsymbol{v}(n)\, \boldsymbol{\alpha}^{*\top}\, \boldsymbol{x}(n)}_{\boldsymbol{u}_{11}} \\
&+ \underbrace{\boldsymbol{v}^\top(n)\, \boldsymbol{x}(n)\, \boldsymbol{v}(n)\, \boldsymbol{v}(n)^\top\, \boldsymbol{x}(n)}_{\boldsymbol{u}_{12}} \Big)
\end{aligned}$$

Let us now analyze $\boldsymbol{v}(n)$ in terms of its expected value. Considering that $z(n)$ is a zero-mean white noise, and using the independence of $\boldsymbol{v}(n)$ and $z(n)$, yields

$$E[\boldsymbol{u}_1] = E[\boldsymbol{u}_2] = E[\boldsymbol{u}_5] = E[\boldsymbol{u}_6] = E[\boldsymbol{u}_7] = E[\boldsymbol{u}_8] = \boldsymbol{0}.$$

After careful calculation, the expected values of the other terms are given by

$$\begin{aligned}
E[\boldsymbol{u}_3] &= c_0\, \mathrm{diag}\{\boldsymbol{R}_x\, \boldsymbol{K}(n)\} \\
E[\boldsymbol{u}_4] &= c_0\, \mathrm{diag}\{\boldsymbol{\alpha}^*\}\, \boldsymbol{R}_x\, E[\boldsymbol{v}(n)] \\
E[\boldsymbol{u}_9] &= \boldsymbol{\alpha}^*\, \boldsymbol{\alpha}^{*\top}\, \boldsymbol{R}_x\, E[\boldsymbol{v}(n)] \\
E[\boldsymbol{u}_{10}] &= \boldsymbol{1}^\top\, [\boldsymbol{R}_x \otimes \boldsymbol{K}(n)]\, \boldsymbol{1}\, \boldsymbol{\alpha}^* \\
E[\boldsymbol{u}_{11}] &= \boldsymbol{K}(n)\, \boldsymbol{R}_x\, \boldsymbol{\alpha}^* \\
E[\boldsymbol{u}_{12}] &= \boldsymbol{K}(n)\, \boldsymbol{R}_x\, E[\boldsymbol{v}(n)]
\end{aligned}$$

where $\boldsymbol{K}(n) = E[\boldsymbol{v}(n)\, \boldsymbol{v}^\top(n)]$ is the covariance matrix of the weight-error vector, and $\otimes$ denotes the Hadamard product. It can be noticed that some of the above expected values require second-moments defined by $\boldsymbol{K}(n)$ in order to update the first-order one $E[\boldsymbol{v}(n)]$. To simplify the analysis, we use the following separation approximation

$$\boldsymbol{K}(n) \approx E[\boldsymbol{v}(n)]\, E[\boldsymbol{v}^\top(n)].$$

This simplification allows us to study, in an analytical way, the mean-weight behavior of the algorithm. See [9] for a thorough discussion on this approximation. Extensive simulation results have shown us that this approximation provides sufficient accuracy in modeling the mean behavior of the weights. We obtain

$$\begin{aligned}
E[\boldsymbol{v}(n+1)] &= E[\boldsymbol{v}(n)] + \ldots \\
&+ \mu\Big(- c_0\, \mathrm{diag}\{\boldsymbol{R}_x\, \boldsymbol{K}(n)\} \\
&- c_0\, \mathrm{diag}\{\boldsymbol{\alpha}^*\}\, \boldsymbol{R}_x\, E[\boldsymbol{v}(n)] \\
&+ \boldsymbol{\alpha}^*\, \boldsymbol{\alpha}^{*\top}\, \boldsymbol{R}_x\, E[\boldsymbol{v}(n)] + \boldsymbol{1}^\top\, [\boldsymbol{R}_x \otimes \boldsymbol{K}(n)]\, \boldsymbol{1}\, \boldsymbol{\alpha}^* \\
&+ \boldsymbol{K}(n)\, \boldsymbol{R}_x\, \boldsymbol{\alpha}^* + \boldsymbol{K}(n)\, \boldsymbol{R}_x\, E[\boldsymbol{v}(n)] \Big)
\end{aligned} \tag{16}$$

The mean-weight behavior $E[\boldsymbol{\alpha}(n)]$ is then obtained by

$$E[\boldsymbol{\alpha}(n)] = \boldsymbol{\alpha}^* + E[\boldsymbol{v}(n)]. \tag{17}$$

Simulations will be used to evaluate the performance of this model.

5. EXPERIMENTAL RESULTS

This section presents simulation examples to illustrate the performance of the proposed algorithm, and to verify the validity of the above analysis. An application to the supervised linear unmixing problem of hyperspectral images is described.

5.1 Algorithm behavior illustration

Consider the identification problem in Figure 1, using the least-mean-square criterion. To illustrate the behavior of the proposed algorithm, the impulse response of the system of reference $\boldsymbol{\alpha}^*$, i.e., the solution of the unconstrained problem (1), was set to

$$\boldsymbol{\alpha}^* = [0.4 \;\; -0.1 \;\; 0.3 \;\; 0.05 \;\; 0 \;\; 0.01 \;\; 0.2 \;\; -0.01 \;\; 0.03 \;\; 0.1]^\top$$

Note that $\boldsymbol{\alpha}^*$ has two negative entries used to activate the non-negativity constraints, and $\|\boldsymbol{\alpha}^*\|_1 = 1.2$. In all the experiments, the same initial vector $\boldsymbol{\alpha}(0) > 0$ was used, normalized so that the sum of its entries is equal to 1. In the first experiment, the inputs $x(n)$ and the noise $z(n)$ were chosen i.i.d. and drawn from a zero-mean Gaussian distribution with variance $\sigma_x^2 = 1$ and $\sigma_z^2 = 0.01$. The non-negativity of the weight coefficients was imposed, and their sum to be equal to 1. The update rule (14) was tested with the step-size values $\mu = 0.05$ and $\mu = 0.15$. The results were averaged over 200 Monte-Carlo runs. Figure 2, first row, shows the learning curves and the mean-weight behavior of the algorithm. One can observe that the algorithm exhibits the very usual behavior of LMS-style approaches, for which small step-size usually means slower convergence but better asymptotic performance. One can also notice that the simulated and theoretical curves representing

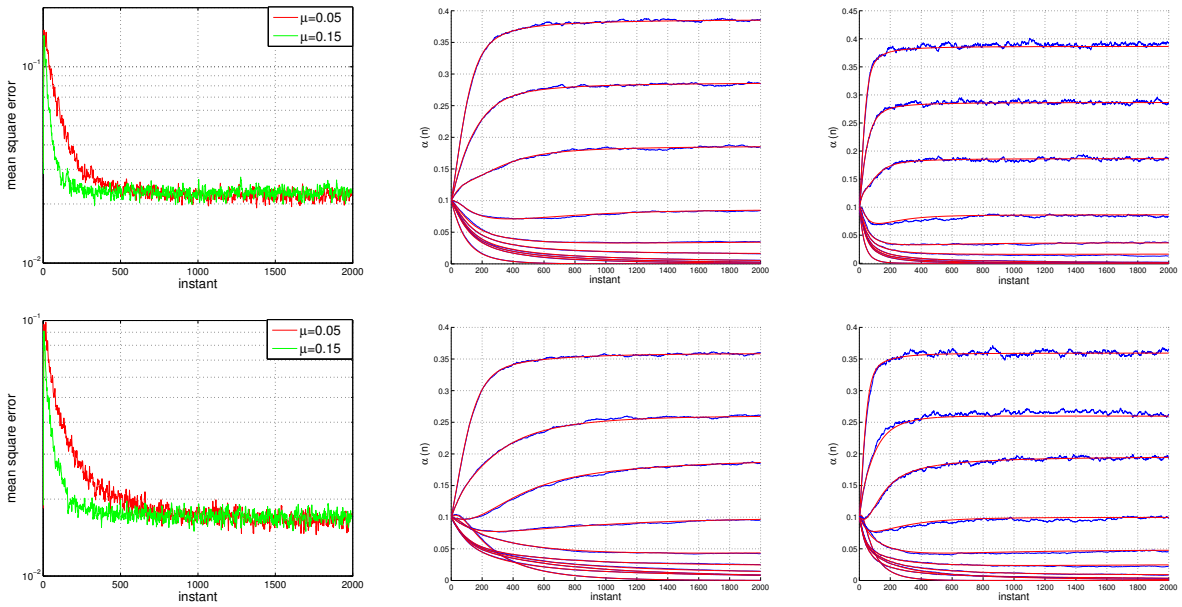

Figure 2: First and second row: results for uncorrelated and correlated input signals, respectively. Left: learning curves; middle and right: Estimated coefficients $\alpha_i(n)$ for $\mu = 0.05$ and $\mu = 0.15$, respectively. The red curves represent the model output (17), and the blue curves are averaged Monte-Carlo runs.

the evolution of $E\{\boldsymbol{\alpha}(n)\}$ are perfectly superimposed. Finally, one can check that the non-negativity and the full-additivity constraints are satisfied.

In the second experiment, our algorithm was tested with correlated inputs $x(n)$. We considered an AR model given by $x(n) = ax(n-1) + w(n)$, with $a = 0.5$. The noise $w(n)$ was chosen i.i.d and zero-mean Gaussian with variance $\sigma_w^2 = 1 - 0.5^2$, so that $\sigma_x^2 = 1$ as the first example. The other parameters remained unchanged. As shown in Figure 2, second row, the same conclusions as above can be repeated.

5.2 Comparison with another method

The performance of the proposed algorithm was also analyzed in the case where $\boldsymbol{\alpha}^*$ is sparse, and compared with the popular projected-gradient scheme. The same experimental setup as above was considered, except that

$$\boldsymbol{\alpha}^* = [0.4 \ -0.1 \ \boldsymbol{O}_{20} \ 0.3 \ 0.05 \ 0 \ \boldsymbol{O}_{30} \ 0.01 \ 0.2 \ \ldots \\ \ldots \ -0.01 \ \boldsymbol{O}_{40} \ 0.03 \ 0.1]^\top$$

where $\boldsymbol{O}_M$ is an all-zero vector of length M. Vector $\boldsymbol{\alpha}^*$ was thus containing 100 entries, among which 90 were null. The projected-gradient method described in [5] was considered for comparison purposes. The authors solve the problem (1)-(3) using the update rule

$$\boldsymbol{\alpha}(n+1) = \Pi(\boldsymbol{\alpha}(n) - \eta \nabla J(\boldsymbol{\alpha}(n)))$$

where Π is the Euclidean projection onto the non-negative ℓ_1 ball. Figure 3(a) compares the learning curves of these two online methods for uncorrelated inputs. Clearly, the proposed method outperformed the projected-gradient method, both in convergence speed and steady-state error. Figures 3(b)-(c) represent the coefficient vector $\boldsymbol{\alpha}(n)$ estimated by the two methods. One can notice that the zeros in $\boldsymbol{\alpha}^*$ were more accurately recovered by our algorithm, because zero is a fixed-point for the update equation (14), at considerably lower computational complexity. Similar results were obtained with correlated inputs.

5.3 Application to hyperspectral data

Our algorithm was used for supervised hyperspectral data unmixing. A key aspect in this problem is to determine abundances (fractions) of different materials under non-negativity and sum-to-one constraints. The studied image is the scene over Moffett Field (CA, USA), captured by the airborne visible infrared imaging spectrometer (AVIRIS). A sub-image of size 50×50 pixels was chosen to evaluate the proposed algorithm. This scene is mainly composed of water, vegetation and soil. The spectral signatures of constituent materials were extracted by the VCA algorithm. The linear unmixing problem can be formulated as

$$\boldsymbol{\alpha}^o = \arg\min_{\boldsymbol{\alpha}} (\boldsymbol{r} - \boldsymbol{M}\boldsymbol{\alpha})^\top (\boldsymbol{r} - \boldsymbol{M}\boldsymbol{\alpha}) \\ \text{subject to} \quad \alpha_i \geq 0, \qquad i = 1, \ldots, L \\ \sum_{i=1}^{L} \alpha_i = 1$$

where $\boldsymbol{\alpha}$ is the vector of abundances, $\boldsymbol{M}$ the $(P \times L)$ matrix of column-wise spectral signatures of constituent materials, and $\boldsymbol{r}$ the observed spectral signature to be unmixed. For this problem, using algorithm (12), we obtained the estimated abundance maps presented in Figure 4. Note that several areas with dominant constituent materials were clearly recovered. These maps

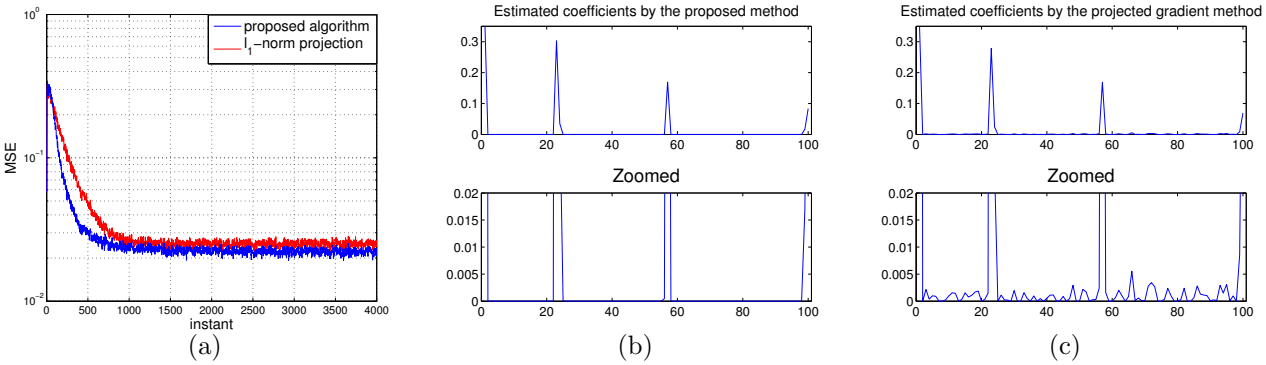

Figure 3: Comparison between the proposed method and the projected-gradient method. (a) Learning curves. (b) Estimated coefficients $\alpha_i(n)$ by the proposed method. (c) Estimated coefficients by the projected-gradient method.

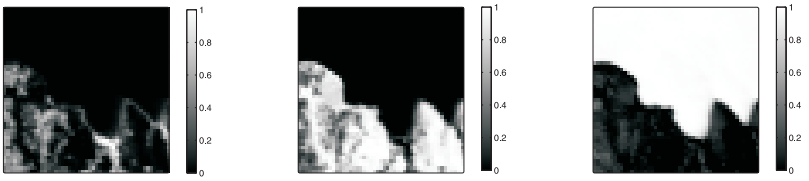

Figure 4: Abundance maps estimated by the proposed algorithm (vegetation, soil and water, respectively)

are very close to those that would be obtained with active set techniques [10] or Bayesian methods [4], which are of a higher level complexity.

6. CONCLUSION

In this paper, we proposed a method for system identification subject to non-negativity and full-additivity constraints. It reaches an optimal solution by traversing the interior of the feasible region, without any extra projection. It achieved faster convergence and lower steady-state error than a projected-gradient based approach, with significantly less computational complexity. We also studied the mean-weight convergence behavior. Future work will focus on analyzing the convergence of the algorithm in the least-squares sense, and obtaining conditions for convergence.

REFERENCES

[1] E. J. Candes and M. Wakin. An introduction to compressive sensing. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

[2] J. Chen, C. Richard, P. Honeine, H. Lantéri, and C. Theys. System identification under non-negativity constraints. In *Proc. EUSIPCO*, Aalborg, Denmark, 2010.

[3] H. Lantéri, M. Roche, O. Cuevas, and C. Aime. A general method to devise maximum-likelihood signal restoration multiplicative algorithms with non-negativity constraints. *Signal Processing*, 81(5):945–974, 2001.

[4] N. Dobigeon, S. Moussaoui, J.-Y. Tourneret, and C. Carteret. Bayesian separation of spectral sources under non-negativity and full additivity constraints. *Signal Processing*, 89(12):2657–2669, 2009.

[5] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l_1-ball for learning in high dimensions. In *Proc. of the 25th international conference on Machine learning*, pages 272–279. ACM, 2008.

[6] S. Theodoridis, K. Slavakis, and I. Yamada. Adaptive learning in a world of projections. *IEEE Signal Processing Magazine*, 28(1):97 –123, 2011.

[7] Y.N. Rao, S.P. Kim, J.C. Sanchez, D. Erdogmus, J.C. Principe, J.M. Carmena, M.A. Lebedev, and M.A. Nicolelis. Learning mappings in brain machine interfaces with echo state networks. In *Proc. IEEE ICASSP*, 2005.

[8] J. Chen, C. Richard, J.-C. M. Bermudez, and P. Honeine. Non-negative least-mean-square algorithm. *IEEE Transactions on Signal Processing*, 2010 (submitted).

[9] P.I. Hubscher and J.-C. M. Bermudez. An improved statistical analysis of the least mean fourth (LMF) adaptive algorithm. *IEEE Transactions on Signal Processing*, 51(3):664 – 671, March 2003.

[10] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. Society for Industrial and Applied Mathematics, 1995.