AN ADAPTIVE GREEDY ALGORITHM WITH APPLICATION TO SPARSE NARMA IDENTIFICATION

Gerasimos Mileounis[#,a], *Behtash Babadi*[*,b], *Nicholas Kalouptsidis*[#,c] *and Vahid Tarokh*[*,d]

[#]*Dept. of Informatics and Telecommunications, University of Athens, Ilisia 15784, Greece*
[*]*School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138*
Email: {gmil[a], kalou[c]}@di.uoa.gr, {behtash[b], vahid[d]}@seas.harvard.edu

ABSTRACT

Greedy algorithms form an essential tool for compressed sensing. However, their inherent batch mode discourages their use in time-varying environments due to significant complexity and storage requirements. In this paper a powerful greedy scheme developed in [1, 2] is converted into an adaptive algorithm which is applied to estimation of nonlinear channels. Performance is assessed via computer simulations on a variety of linear and nonlinear channels; all confirm significant improvements over conventional methods.

Index Terms— Adaptive filters, ARMA processes, Nonlinear systems, Compressed sensing

I. INTRODUCTION

Many real-life systems admit sparse representations, that is they are characterized by small number of non-zero coefficients. Sparse systems can be found in many signal processing applications [3].

Two major algorithmic approaches to compressive sensing are ℓ_1-minimization (basis pursuit) and greedy algorithms (matching pursuit). Basis pursuit methods solve a convex minimization problem, which replaces the ℓ_0 quasi-norm by the ℓ_1 norm. The convex minimization problem can be solved using linear programming methods, and is thus executed in polynomial time [4]. Greedy algorithms, on the other hand, iteratively compute the support set of the signal and construct an approximation to it, until a halting condition is met [1, 2, 5]. Both of the above approaches pose their own advantages and disadvantages. ℓ_1-minimization methods provide theoretical performance guarantees but they lack the speed of greedy techniques. Recently developed greedy algorithms, such as those developed in [1, 2], deliver some of the same guarantee as ℓ_1-minimization approaches with less computational cost and storage.

Many signal processing applications require adaptive estimation with minimal complexity and small memory requirements. Existing approaches to sparse adaptive estimation use the ℓ_1-minimization technique, in order to improve the performance of conventional algorithms. Chen et al. [6] incorporated two different sparsity constraints (the ℓ_1 and the log-sum penalty functions) into the quadratic cost of the standard Least Mean Squares (LMS) to improve the filtering performance on sparse systems. In [7], Angelosante et al. developed a recursive subgradient-based approach for solving the batch Lasso estimator. An ℓ_1-regularized RLS type algorithm based on a low complexity Expectation-Maximization is derived in [8] by Babadi et al. Sparse adaptive ℓ_1-regularized algorithms based on Kalman filtering and Expectation Maximization are reported in [9] by Kalouptsidis et al.

In contrast to the above work on adaptive sparse identification, this paper focuses on the greedy viewpoint. Greedy algorithms, in their ordinary mode of operation, have an inherent batch mode and hence are not suitable for time-varying environments. This paper establishes a conversion procedure that turns greedy algorithms into adaptive schemes for sparse system identification. In particular, a Sparse Adaptive Orthogonal Matching Pursuit (SpAdOMP) algorithm of linear complexity is developed, based on existing greedy algorithms [1, 2] that provide optimal performance guarantees. The developed algorithm is used to estimate ARMA and Nonlinear ARMA channels. Computer simulations show that the proposed algorithm outperforms most existing adaptive algorithms for sparse channel estimation.

The paper is structured as follows. The problem formulation and literature review are addressed in section II. Section III describes the established algorithm along with applications to nonlinear sparse channels. Computer simulations are presented in section IV. Conclusions are discussed in section V.

II. GREEDY METHODS AND THE COSAMP ALGORITHM

Consider the noisy representation of a vector $\boldsymbol{y}(n) = [y_1, \cdots, y_n]^T$ in terms of a basis arranged in the columns of a matrix $\boldsymbol{\Phi}(n)$ at time n

$$\boldsymbol{y}(n) = \boldsymbol{\Phi}(n)\boldsymbol{c} + \boldsymbol{\eta}(n) \tag{1}$$

where $\boldsymbol{c}$ is the parameter vector, $\boldsymbol{\Phi}(n) = [\boldsymbol{\phi}(1), \ldots, \boldsymbol{\phi}(n)]^T$ and $\boldsymbol{\eta}(n) = [\eta_1, \ldots, \eta_n]^T$ is the additive noise. The measurement matrix $\boldsymbol{\Phi}(n) \in \mathbb{C}^{n\times N}$ is often referred to as *dictionary* and the parameter vector $\boldsymbol{c}$ is assumed to be sparse, *i.e.*, $\|\boldsymbol{c}\|_{\ell_0} \ll N$, where $\|\cdot\|_{\ell_0} = |\text{supp}(\cdot)|$ is the ℓ_0 quasi-norm.

We will call the parameter vector s-*sparse* when it contains at most s non-zero entries.

Recovery of the unknown parameter vector $\boldsymbol{c}$ can be pursued by finding the sparsest estimate of $\boldsymbol{c}$ which satisfies the ℓ_2 norm error tolerance δ

$$\min_{\boldsymbol{c}} \|\boldsymbol{c}\|_{\ell_0} \quad \text{subject to} \quad \|\boldsymbol{y}(n) - \boldsymbol{\Phi}(n)\boldsymbol{c}\|_{\ell_2} \leq \delta. \quad (P_{\ell_0})$$

Convex relaxation methods cope with the intractability of the above formulation by approximating the ℓ_0 quasi-norm by the convex ℓ_1 norm. The set of resulting techniques is referred to as ℓ_1-minimization. The ℓ_2 constraint can be interpreted as a noise removal mechanism with $\delta \geq \|\boldsymbol{\eta}(n)\|_{\ell_2}^2$.

The exact conditions for retrieving the sparse vector rely either on the coherence of the measurement matrix [5] or on the Restricted Isometry Property (RIP) [4]. A measurement matrix $\boldsymbol{\Phi}(n)$ satisfies the Restricted Isometry Property for $\delta_s \in (0, 1)$ if we have

$$(1 - \delta_s(n))\|\boldsymbol{c}\|_{\ell_2}^2 \leq \|\boldsymbol{\Phi}(n)\boldsymbol{c}\|_{\ell_2}^2 \leq (1 + \delta_s(n))\|\boldsymbol{c}\|_{\ell_2}^2. \quad (2)$$

for all s-*sparse* $\boldsymbol{c}$. When δ is small, the restricted isometry property implies that the set of columns of $\boldsymbol{\Phi}(n)$ approximately form an orthonormal system.

A. The CoSaMP greedy algorithm

Greedy algorithms provide an alternative approach to ℓ_1-minimization. For the recovery of a sparse signal in the presence of noise, greedy algorithms iteratively improve the current estimate for the parameter vector $\boldsymbol{c}$ by modifying one or more parameters until a halting condition is met. The basic principle behind greedy algorithms is to iteratively find the support set of the sparse vector and reconstruct the signal using the restricted support Least Squares (LS) estimate. The computational complexity of these algorithms depends on the number of iterations required to find the correct support set. One of the earliest algorithms proposed for sparse signal recovery is the Orthogonal Matching Pursuit (OMP) [5]. At each iteration, OMP finds the column of $\boldsymbol{\Phi}(n)$ most correlated with the residual, $\boldsymbol{v}(n) = \boldsymbol{y}(n) - \boldsymbol{\Phi}(n)\hat{\boldsymbol{c}}$, using the proxy signal $\boldsymbol{p}(n) = \boldsymbol{\Phi}^{*T}(n)\boldsymbol{v}(n)$, and adds it to the support set. Then, it solves the following least squares problem:

$$\hat{\boldsymbol{c}} = \arg\min_{\boldsymbol{z}} \|\boldsymbol{y}(n) - \boldsymbol{\Phi}(n)\boldsymbol{z}\|_{\ell_2}$$

and finally updates the residual by removing the contribution of the latter column. By repeating this procedure a total of s times, the support set of $\boldsymbol{c}$ is recovered. Although OMP is quite fast, it is unknown whether it succeeds on noisy measurements.

A more sophisticated algorithm, called Compressed Sampling Matching Pursuit algorithm (CoSaMP) and developed by Needell and Tropp [1], is known to provide nearly optimal performance guarantees. An algorithm similar to the CoSaMP, was presented by Dai and Milenkovic and is known as Subspace Pursuit (SP) [2].

As with most greedy algorithms, CoSaMP takes advantage of the measurement matrix $\boldsymbol{\Phi}(n)$ which is approximately orthonormal ($\boldsymbol{\Phi}^{*T}(n)\boldsymbol{\Phi}(n)$ is close to the identity). Hence, the largest components of the signal proxy $\boldsymbol{p}(n) = \boldsymbol{\Phi}^{*T}(n)\boldsymbol{\Phi}(n)\boldsymbol{c}$ most likely correspond to the nonzero entries of $\boldsymbol{c}$. Next, the algorithm adds the largest components of the signal proxy to the running support set and performs least squares to get an estimate for the signal. Finally, it prunes the least square estimate and updates the error residual. The main ingredients of the CoSaMP algorithm are outlined below:

1. *Identification* of the largest $2s$ components of the proxy signal
2. *Support Merger*, which forms the union of the set of newly identified components with the set of indices corresponding to the s largest components of the least square estimate obtained in the previous iteration
3. *Estimation* via least squares on the merged set of components
4. *Pruning*, which restricts the LS estimate to its s largest components
5. *Sample update*, which updates the error residual.

The above steps are repeated until a halting criterion is met. The main difference between CoSaMP and SP is in the identification step where the SP algorithm chooses the s largest components.

In a time-varying environment, the estimates must be updated adaptively to take into consideration system variations. In such cases, the use of existing greedy algorithms on a measurement block requires that the system remain constant within that block. Moreover, the cost of repetitively applying a greedy algorithm after a new block arrives becomes enormous. Adaptive algorithms, on the other hand, allow online operation. Therefore, our primary goal is to convert existing greedy algorithms into an adaptive mode, while maintaining their superior performance gains. We demonstrate below that the conversion is feasible with linear complexity. We focus our analysis on CoSaMP/SP due to its superior performance, but similar ideas are applicable to other greedy algorithms as well.

III. SPARSE ADAPTIVE ORTHOGONAL MATCHING PURSUIT ALGORITHM

This section starts by converting CoSaMP and SP algorithms [1, 2] into an adaptive scheme. The derived algorithm is then used to estimate sparse Nonlinear ARMA channels.

The proposed algorithm relies on three modifications to the CoSaMP/SP structure: the proxy identification, estimation, and error residual update. The error residual is now evaluated by

$$v(n) = y(n) - \boldsymbol{\phi}^T(n)\boldsymbol{c}(n). \quad (3)$$

The above formula involves the current sample only, in contrast to the CoSaMP scheme which requires all the previous

Table 1. The SpAdOMP Algorithm

Algorithm description		Complexity
$\boldsymbol{p}(0) = 0, \boldsymbol{c}(0) = 0, \boldsymbol{w}(0) = 0$	{Initiliazation}	
$v(0) = y(0)$	{Initial residual}	
$0 < \mu < 2\lambda_{\max}^{-1}$	{Step size}	
For $n := 1, 2, \ldots$ **do**		
$\boldsymbol{p}(n) = \boldsymbol{p}(n-1) + \boldsymbol{\phi}^*(n-1)v(n-1)$	{Form signal proxy}	N
$\Omega = \mathrm{supp}(\boldsymbol{p}_{2s}(n))$	{Identify large components}	N
$\Lambda = \Omega \cup \mathrm{supp}(\boldsymbol{c}(n-1))$	{Merge supports}	s
$\varepsilon(n) = y(n) - \boldsymbol{\phi}_{\|\Lambda}^T(n)\boldsymbol{w}_{\|\Lambda}(n-1)$	{Prediction error}	s
$\boldsymbol{w}_{\|\Lambda}(n) = \boldsymbol{w}_{\|\Lambda}(n-1) + \mu\boldsymbol{\phi}_{\|\Lambda}^*(n)\varepsilon(n)$	{LMS iteration}	s
$\Lambda_s = \max(\|\boldsymbol{w}_{\|\Lambda}(n)\|, s)$	{Obtain the pruned support}	s
$\boldsymbol{c}_{\|\Lambda_s}(n) = \boldsymbol{w}_{\|\Lambda_s}(n), \quad \boldsymbol{c}_{\|\Lambda_s^c}(n) = \mathbf{0}$	{Prune the LMS estimates}	
$v(n) = y(n) - \boldsymbol{\phi}^T(n)\boldsymbol{c}(n)$	{Update error residual}	s
end For		$\mathcal{O}(N)$

samples. Eq. (3) requires s complex multiplications, whereas the cost of the sample update in the CoSaMP/SP is sn multiplications. A new proxy signal that is more suitable for the adaptive mode, can defined as:

$$\boldsymbol{p}(n) = \sum_{i=1}^{n-1} \boldsymbol{\phi}^*(i)v(i) \tag{4}$$

and is updated by $\boldsymbol{p}(n) = \boldsymbol{p}(n-1) + \boldsymbol{\phi}^*(n-1)v(n-1)$. The last modification attacks the estimation step. The vector $\boldsymbol{c}$ is updated by standard adaptive algorithms such as the LMS and RLS.

LMS is one of the most widely used algorithm in adaptive filtering due to its simplicity, robustness and low complexity. Hence, for reasons of simplicity and complexity we focus on the LMS algorithm. At each iteration the current regressor $\boldsymbol{\phi}(n)$ and the previous estimate $\boldsymbol{w}(n-1)$ are restricted to the instantaneous support originated from the support merging step. The resulting algorithm is presented in Table 1, where $\boldsymbol{\phi}_{|\Lambda}$ and $\boldsymbol{w}_{|\Lambda}$ denote the sub-vectors corresponding to the index set Λ, $\max(|a|, s)$ returns s indices of the largest elements of a and Λ^c represents the complement of set Λ. The following Theorem establishes the steady state Mean Square Error (MSE) error performance of the SpAdOMP algorithm:

Theorem 1. *(SpAdOMP)*[1]*. The proposed algorithm, for large n, produces an s-sparse approximation $\boldsymbol{c}(n)$ that satisfies the following steady-state error bound*

$$\|\boldsymbol{c} - \boldsymbol{c}(n)\|_{\ell_2} \lesssim C_1(n)\|\boldsymbol{\eta}(n)\|_{\ell_2} + C_2(n)\|\boldsymbol{\phi}_{|\Lambda}(n)\|_{\ell_2}|e_o(n)|,$$

where $e_o(n)$ is the estimation error of the optimum Wiener filter and $C_1(n)$, $C_2(n)$ are constants independent of $\boldsymbol{c}$ (given explicitly in [10]) and are only functions of the restricted isometry constants, λ_{min} (the minimum eigenvalue of the input covariance matrix) and the step-size μ.

[1]Proof is omitted due to space limitations.

A. Sparse NARMA identification

The nonlinear model that we will be concerned with, constitutes a generalization of the class of linear ARMA models [11] and is known as Nonlinear AutoRegressive Moving Average (NARMA) [12]. The output of NARMA models depends on past and present values of the input as well as the output

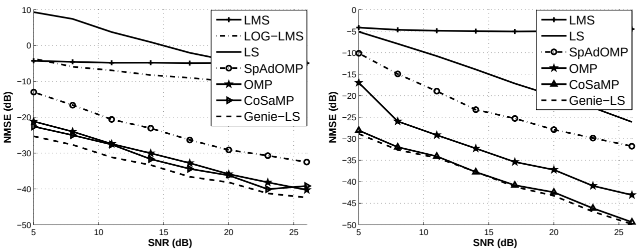
$$y_i = f(y_{i-1}, \ldots, y_{i-M_y}, x_i, \ldots, x_{i-M_x}) + \eta_i \tag{5}$$

where y_i, x_i and η_i are the system output, input and noise, respectively; M_y, M_x denote the output and input memory orders; η_i is Gaussian and independent from x_i; and $f(.)$ is a sparse polynomial function in several variables with degree of nonlinearity p. Known linearization criteria [11] provide sufficient conditions for the Bounded Input Bounded Output stability of (5).

Using Kronecker products we write Eq. (5) as a linear regression model

$$y_i = \boldsymbol{\phi}^T(i)\boldsymbol{c} + \eta_i \tag{6}$$

where $\boldsymbol{y}_i = \left[y_{i-1}, \cdots, y_{i-M_y}\right]^T$, $\boldsymbol{x}_i = [x_i, \cdots, x_{i-M_x}]^T$ and $\boldsymbol{\phi}^T(i) = \left[\ \boldsymbol{\phi}_y^T(i) \quad \boldsymbol{\phi}_x^T(i) \quad \boldsymbol{\phi}_{yx}^T(i)\ \right]$. Consider the p-fold Kronecker products $\boldsymbol{y}_i^{(p)} = [\otimes_{j=1}^p \boldsymbol{y}_i]$ and $\boldsymbol{x}_i^{(p)} = [\otimes_{j=1}^p \boldsymbol{x}_i]$, then the output and input regressor vectors are respectively given by $\boldsymbol{\phi}_y^T(i) = [\boldsymbol{y}_i^{(1)}, \cdots, \boldsymbol{y}_i^{(p)}]$ and $\boldsymbol{\phi}_x^T(i) = [\boldsymbol{x}_i^{(1)}, \cdots, \boldsymbol{x}_i^{(p)}]$. $\boldsymbol{\phi}_{yx}^T(i)$ denotes all possible Kronecker product combinations of $\boldsymbol{y}_i$ and $\boldsymbol{x}_i$ of degree up to p. The components of $\boldsymbol{c} = [\ \boldsymbol{c}_y^T \quad \boldsymbol{c}_x^T \quad \boldsymbol{c}_{yx}^T]^T$ correspond to the coefficients of the polynomial $f(\cdot)$. Hence, if we collect n



(a) ARMA channel (b) NARMA channel

Fig. 1. NMSE of the channel estimates versus SNR

successive observations, recovery of the sparsest parameter vector can be accomplished by solving the mathematical program (P_{ℓ_0}).

IV. SIMULATION RESULTS

In this section, we compare through computer simulations the performance of existing algorithms and the algorithm proposed in this paper. Experiments were conducted on both linear and nonlinear channel setups. In all experiments the output sequence is disturbed by additive white Gaussian noise for various SNR levels ranging from 5 to $26dB$. SNR is the ratio of the noiseless channel output power to the noise power corrupting the output signal $(\sigma_y^2/\sigma_\eta^2)$. The Normalized Mean Square Error, defined as $\mathbb{E}[\|\boldsymbol{c}(n) - \boldsymbol{c}\|_{\ell_2}^2]/\mathbb{E}[\|\boldsymbol{c}\|_{\ell_2}^2]$ is used to assess performance. To reduce the realization dependency, the parameter estimates were averaged over 30 Monte Carlo runs.

In the first experiment sparse ARMA channel estimation is considered. The channel memory is $M_y = M_x = 50$ and the channel to be estimated is given by

$$y_n = a_1 y_{n-6} + a_2 y_{n-48} + x_n + b_1 x_{n-13} + b_2 x_{n-34}$$

where $[a_1, a_2] = [-0.5167 - j0.2828, 0.1801 + j0.1347]$ and $[b_1, b_2] = [-0.5368 - j0.9198, 1.0719 + j0.0318]$. The input sequence is drawn from a complex Gaussian distribution, $\mathcal{CN}(0, 1/5)$, and the number of samples processed was 500. The step size for the conventional LMS, log-LMS [6] and the SpAdOMP was set to $\mu_{\text{LMS}} = 1 \times 10^{-2}$, $\mu_{\text{LOG-LMS}} = 2 \times 10^{-2}$ and $\mu_{\text{SpAdOMP}} = 7 \times 10^{-2}$. Fig. 1(a) shows the excellent performance match between the Genie LS (which knows the true support), CoSaMP [1] and OMP [5], all of which have quadratic complexity. The LMS, log-LMS and SpAdOMP have an order of magnitude less computational complexity, but only SpAdOMP achieves a performance gain close to Genie LS ($9dB$ less).

In the second experiment the following NARMA channel is considered

$$y_n = a_1 y_{n-50} + a_2 y_{n-9}^2 + b_1 x_{n-8} + b_2 |x_{n-21}|^2 x_{n-21}$$

where $[a_1, a_2] = [-0.1586 - j0.7064, -0.1428 - j0.0478]$ and $[b_1, b_2] = [-0.8082 - j0.5221, -0.5177 + j0.7131]$ and the channel memory is $M_y = M_x = 50$. The input sequence is generated from a complex Gaussian distribution, $\mathcal{CN}(0, 1/4)$, consisting of 1000 samples. The step size parameters $\mu_{LMS} = 6 \times 10^{-3}$ and $\mu_{SpAdOMP} = 0.3$ are used for the conventional LMS and SpAdOMP. OMP and SpAdOMP lag behind Genie LS by $5dB$ and $12dB$ respectively in performance. It is worth pointing out that SpAdOMP obtains an average gain of $19dB$ over the conventional LMS. Note that this significant NMSE gain is the product of both the denoising mechanism and the compressed sampling virtue of the CoSaMP/SP algorithm, which are lacking in conventional adaptive algorithms such as LMS.

V. CONCLUSIONS

In this paper, an adaptive algorithm for sparse approximations with linear complexity was developed using the underlying principles of existing batch-greedy algorithms. The proposed algorithm was applied to sparse NARMA identification. Simulation results validated the superior performance of the new algorithm.

VI. REFERENCES

[1] D. Needell and J.A. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples,” *Appl. Comput. Harmon. Anal*, vol. 26, pp. 301–321, 2009.

[2] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing signal reconstruction,” *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[3] A.M. Bruckstein, D.L. Donoho, and M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.

[4] E. Candés and T. Tao, “Decoding by linear programming,” *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, 2004.

[5] J.A. Tropp, “Greed is good: algorithmic results for sparse approximation,” *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[6] Y. Chen, Y. Gu, and A. Hero, “Sparse LMS for system identification,” in *Proc. IEEE ICASSP*, 2009, pp. 3125–3128.

[7] D. Angelosante and G.B. Giannakis, “RLS-weighted LASSO for adaptive estimation of sparse signals,” in *Proc. IEEE ICASSP*, 2009, pp. 3245–3248.

[8] B. Babadi, N. Kalouptsidis, and V. Tarokh, “SPARLS: The sparse RLS algorithm,” *submitted*, 2009.

[9] N. Kalouptsidis, G. Mileounis, B. Babadi, and V. Tarokh, “Adaptive algorithms for sparse nonlinear channel estimation,” in *Proc. IEEE SSP*, 2009, pp. 221–224.

[10] G. Mileounis, B. Babadi, N. Kalouptsidis, and V. Tarokh, “An adaptive greedy algorithm with application to nonlinear communications,” *submitted*, 2009.

[11] N. Kalouptsidis, *Signal Processing Systems: Theory & Design*, Wiley, 1997.

[12] S. Chen and S.A. Billings, “Representations of non-linear systems: the NARMAX model,” *Int. J. Control*, vol. 49, no. 3, pp. 1013–1032, 1989.