RLS-WEIGHTED LASSO FOR ADAPTIVE ESTIMATION OF SPARSE SIGNALS

Daniele Angelosante

Università degli studi di Cassino
Via Di Biasio, 43
03043 Cassino (FR), Italy

Georgios B. Giannakis *

University of Minnesota
Dept. of ECE, 200 Union Str. SE
Minneapolis, MN 55455, USA

ABSTRACT

The batch least-absolute shrinkage and selection operator (Lasso) has well-documented merits for estimating sparse signals of interest emerging in various applications, where observations adhere to parsimonious linear regression models. To cope with linearly growing complexity and memory requirements that batch Lasso estimators face when processing observations sequentially, the present paper develops a recursive Lasso algorithm that can also track slowly-varying sparse signals of interest. Performance analysis reveals that recursive Lasso can either estimate consistently the sparse signal's support or its nonzero entries, but not both. This motivates the development of a weighted version of the recursive Lasso scheme with weights obtained from the recursive least-squares (RLS) algorithm. The resultant RLS-weighted Lasso algorithm provably estimates sparse signals consistently. Simulated tests compare competing alternatives and corroborate the performance of the novel algorithms in estimating time-invariant and tracking slow-varying signals under sparsity constraints.

***Index Terms*—** Lasso, Variable Selection, Sparsity, Tracking.

1. INTRODUCTION

Sparsity is a feature present in a plethora of natural as well as man-made signals and systems. This is reasonable not only because nature itself is parsimonious but also because simple models and processing with minimal degrees of freedom are attractive from an implementation perspective. Exploitation of sparsity is critical in applications as diverse as variable selection in linear regression models for diabetes [9], image compression [4], and distributed spectrum sensing for cognitive radios [2]. To name a few, sparsity-aware signal estimators include the basis pursuit and Lasso operators, the Dantzig selector and recent ones that appeared with the emerging area of compressive sampling; see [9, 5, 4] and references therein.

The aforementioned estimators entail the ℓ_1 norm of the signal of interest and are nonlinear functions of the available observations, which they process in a *batch* form using iterative linear programming solvers. Recently, recovery of noise-free sparse signals from linear projections taken one at a time until perfect reconstruction, was studied in [8] along with optimal stopping rules and pertinent implementation issues. Many sparse signals encountered in practice however, have to be estimated based on noisy observations that become available *sequentially* in time. For such cases, batch signal estimators typically incur complexity and memory requirements that grow as time progresses. In addition, the sparse signal may vary with time both in its nonzero support set as well as in the values of its nonzero entries.

To cope with these challenges, the present paper develops adaptive algorithms for recursive estimation and tracking of (possibly time-varying) sparse signals based on noisy sequential observations adhering to a linear regression model. The novel schemes outlined in Sections 3 and 4 are termed recursive (R) Lasso and recursive least-squares (RLS) weighted Lasso because they constitute online counterparts of the batch Lasso [9] and the weighted Lasso [11], respectively. When the signal of interest is time-invariant, the performance of R-Lasso and RLS-weighted Lasso is analyzed in Section 4 to assess consistency in estimating the support set as well as the values of the nonzero sparse signal entries. These performance results in the recursive regime complement rather nicely those derived in [10, 11, 12] for batch estimators of sparse signals. Corroborating simulations are presented in Section 5 both for time-invariant and time-varying sparse signals, where comparisons are also drawn among R-Lasso, RLS-weighted Lasso, sparsity-unaware recursive as well as sparsity-aware batch estimators. Conclusions are drawn in Section 6.

*Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

2. PRELIMINARIES AND PROBLEM STATEMENT

An $N \times 1$ vector $\mathbf{x}$ is called sparse if only a few of its entries $\{x_n\}_{n=1}^N$ are nonzero. Upon defining the nonzero support set of $\mathbf{x}$ as $\text{supp}(\mathbf{x}) := \{n \in \{1, \ldots, N\} : x_n \neq 0\}$, sparsity amounts to having $|\text{supp}(\mathbf{x})| \ll N$, where $|\cdot|$ denotes set cardinality. Suppose that such a sparse signal vector $\mathbf{x}$ is to be obtained *sequentially* from $K \times 1$ vector observations $\{\mathbf{y}_\tau\}_{\tau=1}^t$. These observations obey the linear regression model

$$\mathbf{y}_\tau = \mathbf{H}_\tau \mathbf{x} + \mathbf{n}_\tau, \qquad \tau = 1, 2 \ldots, t \tag{1}$$

where $\{\mathbf{H}_\tau\}_{\tau=1}^t$ are known $K \times N$ regression matrices, and the noise vectors $\{\mathbf{n}_\tau\}_{\tau=1}^t$ are assumed zero mean and uncorrelated across time, each with known covariance matrix $\sigma^2 \mathbf{I}_K$.

Given $\{\mathbf{y}_\tau, \mathbf{H}_\tau\}_{\tau=1}^t$ and σ^2, a *batch* approach to estimating the sparse $\mathbf{x}$ is provided by the least-absolute shrinkage and selection operator (Lasso) [9] – a method also known in the signal processing community as basis pursuit denoising [5]. This approach can be readily applied to the concatenated model $\boldsymbol{y}_t = \boldsymbol{\mathcal{H}}_t \mathbf{x} + \boldsymbol{n}_t$, with $\boldsymbol{y}_t := [\mathbf{y}_1^T, \ldots, \mathbf{y}_t^T]^T$ and $\boldsymbol{\mathcal{H}}_t := [\mathbf{H}_1^T, \ldots, \mathbf{H}_t^T]^T$ (^{T} stands for transpose). The batch Lasso estimates the wanted sparse vector as

$$\hat{\mathbf{x}}_t = \arg\min_{\mathbf{x}} \left[\frac{1}{2} \|\boldsymbol{y}_t - \boldsymbol{\mathcal{H}}_t \mathbf{x}\|_2^2 + \lambda_t \|\mathbf{x}\|_1 \right] \tag{2}$$

where the ℓ_2 norm in the objective function denotes the ordinary least-squares (OLS) cost; the ℓ_1 norm $\|\mathbf{x}\|_1 := \sum_{n=1}^{N} |x_n|$ effects the sparsity constraint; and λ_t is a penalty parameter which can be tuned to trade off the OLS error for the number of the nonzero entries (degree of sparsity) in the estimate [9, 5].

Albeit non-differentiable, the Lasso objective is convex in $\mathbf{x}$, and can be optimized using linear programming techniques [3]. Obtaining the global optimum in (2) is thus, in principle, tractable. Unfortunately, the size of $\boldsymbol{y}_t$ and $\boldsymbol{\mathcal{H}}_t$ grows linearly with t in the sequential setup considered here. As a result, batch Lasso solvers that incur complexity in the order of $\mathcal{O}\left(KtN \min(Kt, N)\right)$ per time instant t, soon become prohibitive both in terms of computational complexity as well as in memory requirements. Additional challenges emerge when sparse signal vectors in practice exhibit slow variations both in their nonzero support set as well as in the values of their nonzero entries, as time t progresses.

In response to these challenges, the goal of this paper is to develop sequential and adaptive Lasso estimators with manageable complexity and memory requirements that are also capable of tracking slow variations in the sparse vector of interest $\mathbf{x}$.

3. RECURSIVE LASSO

If the sparsity constraint is not present, the goal of adaptive estimation and tracking slow-varying signals adhering to a linear regression model can be accomplished by the well known recursive least-squares (RLS) algorithm [7, Chap. 8]. RLS yields online the estimate

$$\hat{\mathbf{x}}_t^{RLS} = \arg\min_{\mathbf{x}} \sum_{\tau=1}^{t} \beta^{t-\tau} \|\mathbf{y}_\tau - \mathbf{H}_\tau \mathbf{x}\|_2^2 \qquad (3)$$

where $\beta \in (0, 1]$ is the forgetting factor chosen to window the data employed in forming the estimator, and thus strike a balance between controlling convergence (to the true $\mathbf{x}$ when the latter is time-invariant) and the ability to track slow variations in $\mathbf{x}$. (Recall that when $\mathbf{x}$ is time-invariant, choosing $\beta = 1$ can render RLS equivalent to the batch OLS.)

Motivated by RLS, consider decomposing the Lasso cost in (2) and including the forgetting factor to obtain

$$J_t(\mathbf{x}) = \frac{1}{2} \sum_{\tau=1}^{t} \beta^{t-\tau} \|\mathbf{y}_\tau - \mathbf{H}_\tau \mathbf{x}\|_2^2 + \lambda_t \|\mathbf{x}\|_1 \ . \qquad (4)$$

A recursive Lasso algorithm (referred as R-Lasso) can then be sought to find iteratively the estimate $\hat{\mathbf{x}}_t^{R-Lasso} = \arg\min_{\mathbf{x}} J_t(\mathbf{x})$. To this end, one can rewrite (4) as $J_t(\mathbf{x}) = (a_t + \mathbf{x}^T \mathbf{R}_t \mathbf{x} - 2\mathbf{x}^T \mathbf{r}_t)/2 + \lambda_t \|\mathbf{x}\|_1$, where $a_t := \sum_{\tau=1}^{t} \beta^{t-\tau} \mathbf{y}_\tau^T \mathbf{y}_\tau$, $\mathbf{r}_t := \sum_{\tau=1}^{t} \beta^{t-\tau} \mathbf{H}_\tau^T \mathbf{y}_\tau$ and $\mathbf{R}_t := \sum_{\tau=1}^{t} \beta^{t-\tau} \mathbf{H}_\tau^T \mathbf{H}_\tau$. Again similar to RLS, these quantities can be updated recursively using the iterations

$$\begin{aligned} \mathbf{r}_t &= \beta \mathbf{r}_{t-1} + \mathbf{H}_t^T \mathbf{y}_t \\ \mathbf{R}_t &= \beta \mathbf{R}_{t-1} + \mathbf{H}_t^T \mathbf{H}_t \ . \end{aligned} \qquad (5)$$

Different from RLS however, gradient-based minimization of $J_t(\mathbf{x})$ is impossible because the ℓ_1 norm is non-differentiable. A possible bypass in such cases is offered by subgradient-based iterative minimizers [3, p. 620]. In the present context, the subgradient vector has nth entry given by[1]

$$\{\check{\nabla} J_t(\mathbf{x})\}_n = \begin{cases} \{\nabla L_t(\mathbf{x})\}_n + \lambda_t \text{sign}(x_n), & \text{if } x_n \neq 0 \\ \{\nabla L_t(\mathbf{x})\}_n - \lambda_t, & \text{if } x_n = 0, \{\nabla L_t(\mathbf{x})\}_n > \lambda_t \\ \{\nabla L_t(\mathbf{x})\}_n + \lambda_t, & \text{if } x_n = 0, \{\nabla L_t(\mathbf{x})\}_n < -\lambda_t \\ 0, & \text{if } x_n = 0, -\lambda_t < \{\nabla L_t(\mathbf{x})\}_n < \lambda_t \end{cases}$$

where $L_t(\mathbf{x}) := (\mathbf{x}^T \mathbf{R}_t \mathbf{x} - 2\mathbf{x}^T \mathbf{r}_t)/2$ is the differentiable OLS cost with gradient $\nabla L_t(\mathbf{x}) = \mathbf{R}_t \mathbf{x} - \mathbf{r}_t$. In practice, the condition $x_n = 0$ in the subgradient is replaced by $|x_n| < \delta \ll 1$, where δ is a prescribed constant.

Since (5) enables online updating of ∇L_t (and thus of $\check{\nabla} J_t$), it follows readily that the subgradient iterates (indexed by i)

$$\mathbf{x}_t^{(i+1)} = \mathbf{x}_t^{(i)} - \alpha_i \check{\nabla} J_t(\mathbf{x}_t^{(i)}) \qquad (7)$$

can be recursively updated at affordable complexity which does not increase as time t progresses. A vanishing stepsize of the form $\alpha_i = \alpha/\sqrt{i}$ or $\alpha_i = \alpha/i$ guarantees convergence (as $i \to \infty$) to the global minimum of J_t; $\alpha_i = \alpha$ ensures convergence within a ball whose radius depends on α.

Remark 1. Similar to the least mean-square (LMS) algorithm [7], it is also possible to develop stochastic subgradient solvers of the R-Lasso minimization problem. Those entail instantaneous subgradient updates with the iteration index i in (7) replaced by the time index t. Nonetheless, LMS-like R-Lasso solvers can afford enhanced adaptability as well as minimal complexity and memory requirements at the expense of slower convergence and steady-state error (misadjustment).

Remark 2. Whether $\mathbf{x}_t^{(i)}$ converges pointwise or within a ball to the global minimizer of J_t, it can be proved that the limit of $\hat{\mathbf{x}}_t^{R-Lasso}$ as $t \to \infty$ does not necessarily converge to the true $\mathbf{x}$ even in the time-invariant scenario [1]. This should not be surprising because even the batch Lasso in (2) is not guaranteed to recover the correct support and at the same time estimate the nonzero entries of $\mathbf{x}$ consistently [11, 6]. Such a shortcoming was recently overcome for the batch Lasso by [11, 12], and motivates the novel RLS-weighted Lasso approach outlined next for adaptive estimation and tracking applications involving sparse signals.

4. PERFORMANCE AND RLS-WEIGHTED LASSO

As the nonzero support set is unknown, performance analysis of sparse signal estimators is distinct from and far more challenging than the performance of OLS and RLS estimators. For Lasso estimators in particular, one intuitively expects that performance properties should also depend on the penalty parameter λ_t. The first desirable property of a sparse signal estimator $\hat{\mathbf{x}}_t$ pertains to (strong) support consistency, which requires

$$\lim_{t\to\infty} \text{Prob}\left[\text{supp}(\hat{\mathbf{x}}_t) = \text{supp}(\mathbf{x})\right] = 1 \qquad (8)$$

while the second property demands (weak) estimation (a.k.a. $\sqrt{t}$) consistency, i.e., with $_d$ denoting convergence in distribution

$$\sqrt{t}\left[\mathcal{S}_{\mathbf{x}}\left(\hat{\mathbf{x}}_t\right) - \mathcal{S}_{\mathbf{x}}\left(\mathbf{x}\right)\right] \to_d \mathcal{N}(\mathbf{0}_{|\text{supp}(\mathbf{x})|,1}, \boldsymbol{\Sigma}^{-1}) \qquad (9)$$

where $\mathcal{S}_{\mathbf{x}}(\mathbf{x}') : \mathbb{R}^N \to \mathbb{R}^{|\text{supp}(\mathbf{x})|}$ is an operator selecting only the entries of $\mathbf{x}'$ corresponding to $\text{supp}(\mathbf{x})$ and $\boldsymbol{\Sigma}$ is a positive definite matrix. Because both properties entail the nonzero support set, which is unknown, they are termed *oracle properties* [6, 11, 10].

[1] Proofs are omitted due to space limitations but can be found in [1].

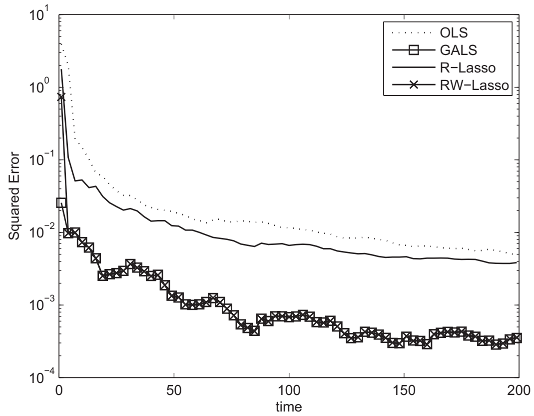

Fig. 1. Squared estimation error when $\mathbf{x}$ is time-invariant.

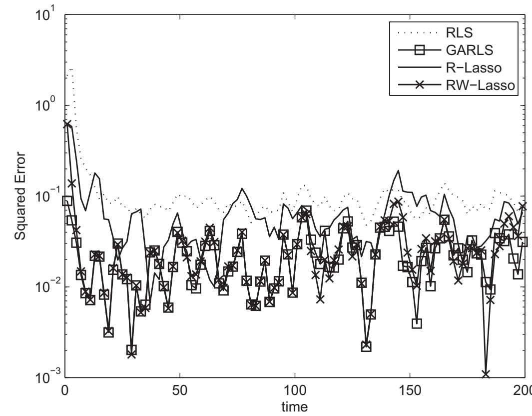

Fig. 2. Squared estimation error when $\mathbf{x}$ is slow-varying.

As alluded to in Remark 2, extending the performance analysis results of [6] for the batch setup, we have proved that the following holds in the sequential regime when $\mathbf{x}$ is time-invariant [1].

Proposition 1. *There exists no λ_t for which R-Lasso can satisfy simultaneously* (8) *and* (9)*. Moreover, if λ_t grows faster than $\sqrt{t}$ but slower than t and the irrepresentable condition in [10] is met, then R-Lasso can guarantee* (8) *but not* (9)*. Finally, with $\lambda_t \propto \sqrt{t}$ the limit in* (8) *is strictly less than one but R-Lasso estimates are at least asymptotically unbiased.*

In search for an alternative to obviate the mostly negative performance results of Proposition 1, it became apparent that one should look for a penalty term that is signal dependent and weighs differently the entries $|x_n|$ in the ℓ_1 norm term in (4). Generalizing the OLS-weighted batch Lasso approach in [11, 12] to the sequential framework herein, we thus came up with the weight function

$$w_{\mu_t}(|x|) := \frac{(a\mu_t - |x|)_+}{\mu_t(a-1)}\, u\,(|x| - \mu_t) \; + \; u\,(\mu_t - |x|) \qquad (10)$$

where $u(\cdot)$ stands for the step function; $(\cdot)_+$ denotes the nonnegative part of the quantity in parentheses; and the parameter a is set to $a = 3.7$ [6]. Using (3) in this weight function, the cost in (4) with unweighted ℓ_1 norm becomes

$$J_t^{RW-Lasso}(\mathbf{x}) \;=\; \frac{1}{2}\sum_{\tau=1}^{t}\beta^{t-\tau}\|\mathbf{y}_\tau - \mathbf{H}_\tau\mathbf{x}\|_2^2 \;+\; \lambda_t \sum_{n=1}^{N} w_{\mu_t}(|\hat{x}_{t,n}^{RLS}|)\,|x_n| \qquad (11)$$

and the resultant RLS-weighted Lasso (RW-Lasso) estimator is given by

$$\hat{\mathbf{x}}_t = \arg\min_{\mathbf{x}} J_t^{RW-Lasso}(\mathbf{x}) \; . \qquad (12)$$

Similar to R-Lasso, the estimator in (12) can be implemented using subgradient iterations. It is slightly more complex than R-Lasso because it requires running in parallel an RLS algorithm to supply the needed weights. In return however, the RLS-weighted Lasso estimator retains the tracking advantages of R-Lasso while for time-invariant $\mathbf{x}$ it can be shown to enjoy the desirable asymptotic performance guarantees, as summarized next [1].

Proposition 2. *If the forgetting factor is chosen as $\beta = 1$ and the penalty parameter λ_t is selected to grow faster than $\sqrt{t}$ but slower than t with $\mu_t = \lambda_t/t$, the RLS-weighted Lasso estimator in* (12) *satisfies the oracle properties* (8) *and* (9)*.*

The ensuing remark discusses performance issues when $\mathbf{x}$ is allowed to be slowly time-varying.

Remark 3. To endow R-Lasso and RW-Lasso with tracking capabilities, β must be chosen less than one. In this case however, neither R-Lasso nor RW-Lasso can provably guarantee the oracle properties. Nonetheless, if the $\mathbf{H}_\tau$ matrices are orthonormal, it is possible to express the two estimates in closed form and prove that RW-Lasso outperforms R-Lasso [1].

5. SIMULATED TESTS

The analytical claims of Sections 3 and 4 are tested here using three simulated examples.

Test Case 1. Gaussian vector observations were generated according to (1) with a time-invariant $\mathbf{x}$ and parameters $N = 100$, $K = 30$, $\sigma^2 = 10^{-2}$ and $\mathbf{n}_\tau \sim \mathcal{N}(\mathbf{0}_{K,1}, \sigma^2\mathbf{I}_K)$ for $\tau = 1, 2, \ldots, 200$. Without loss of generality, the first five entries of $\mathbf{x}$ were chosen equal to unity and all other entries equal to zero; i.e., supp($\mathbf{x}$) $=\{1, 2, 3, 4, 5\}$ and $\mathcal{S}_{\mathbf{x}}(\mathbf{x}) = [1, 1, 1, 1, 1]^T$. Matrix $\mathbf{H}_t$ was formed with entries drawn from a zero-mean Gaussian distribution with variance $1/K$. Setting $\beta = 1$, the simulated algorithms included: (i) the OLS; (ii) the genie-aided (GA)LS, which is the support-aware LS estimator applied to a reduced model after removing the regressors corresponding to the zero entries of $\mathbf{x}$; (iii) the R-Lasso with $\lambda_t = \sqrt{2\sigma^2 t \log N}$ and the RW-Lasso with $\lambda_t = \sqrt{2\sigma^2 t^{4/3}\log N}$ and $\mu_t = \lambda_t/t$. Note that these choices for λ_t and μ_t guarantee that the RW-Lasso satisfies the oracle properties. The R-Lasso and RW-Lasso estimates were obtained iteratively using the subgradient method described in Section 3 with $\alpha_i = \alpha/\sqrt{i}$. Fig. 1 depicts the square-error (SE) of the estimates across time, that is $\text{SE}_t := \|\hat{\mathbf{x}}_t - \mathbf{x}\|_2^2$. Because it exploits sparsity, R-Lasso outperforms the OLS. But it is outperformed by RW-Lasso, whose performance after a certain time approaches that of GALS, thus corroborating the claim that the RW-Lasso satisfies the oracle properties when $\mathbf{x}$ is time-invariant.

Test Case 2. The simulation setup here aimed at estimating a sparse signal $\mathbf{x}_t$ with time-invariant supp($\mathbf{x}_t$) $=\{1, 2, 3, 4, 5\}$,

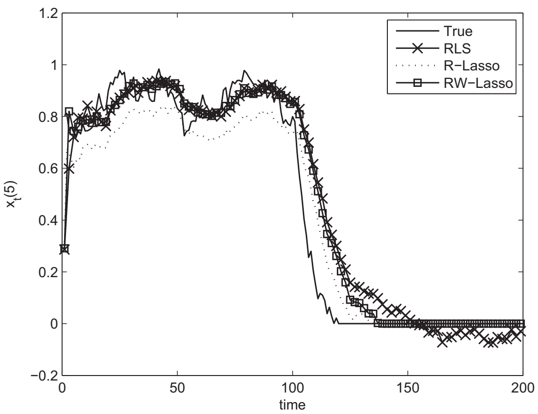

Fig. 3. Time evolution of the x_5 signal entry.

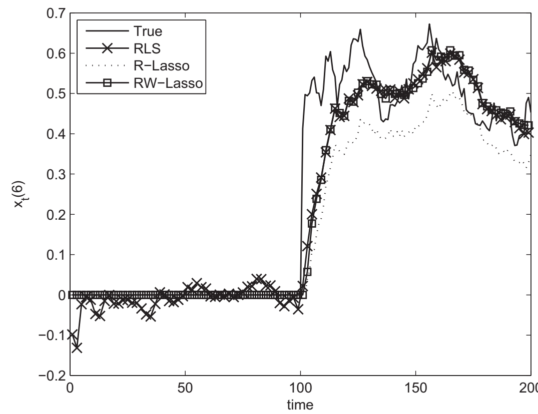

Fig. 4. Time evolution of the x_6 signal entry.

but with slowly-varying nonzero entries, each obeying a first-order Gauss-Markov process $x_{t+1,n} = (1-\rho)x_{t,n} + \sqrt{\rho}z_{t,n}$ with $\rho = 10^{-3}$, $z_{t,n} \sim \mathcal{N}(0,1)$ and initial entries $x_{1,n} \sim \mathcal{N}(0,1)$ for $n = 1,\ldots,5$. All other parameters were selected as in Test Case 1. The same four algorithms were tested but now with $\beta = 0.9$, $\lambda_t = \sqrt{2\sigma^2 \log N}\sqrt{\sum_{\tau=1}^{t} \beta^{2(t-\tau)}}$ and $\mu_t = \lambda_t / \sum_{\tau=1}^{t} \beta^{t-\tau}$. The last choices were made because they exhibited reliable performance with orthonormal regression matrices where R-Lasso and RW-Lasso estimates are available in closed form (cf. Remark 3). Fig. 2 confirms that even in a slow-varying setup the R-Lasso outperforms the RLS and is outperformed by the RW-Lasso. Since $\beta < 1$ here, there is no hope to satisfy the oracle properties. Nevertheless, the performance of RW-Lasso again comes close to that of GARLS, which is the RLS matched to the true support.

Test Case 3. The setting in this example is identical to that of Test Case 2, except that here the support of the sparse signal also undergoes step changes. Specifically, at $t = 100$ the fifth entry of $\mathbf{x}$ starts decreasing while after $t = 120$ the same entry is set to zero. In addition, at $t = 100$ the sixth entries becomes nonzero. Figs. 3 and 4 depict, respectively, the true variations of x_5 and x_6 across time, along with the RLS, the R-Lasso and the RW-Lasso estimated trajectories. As expected, the RLS estimates are not sparse and can assume a nonzero value even if the true entry is zero; while the R-Lasso estimates are sparse but clearly under-estimate the true signal variations. RW-Lasso estimates on the other hand, remain close to RLS ones for the nonzero entries but succeed also in bringing the null entries of the sparse signal close to zero, as they should.

6. CONCLUSIONS

Adaptive algorithms were developed in this paper for recursive estimation and tracking of (possibly time-varying) sparse signals based on observations that obey a linear regression model and become available sequentially in time. A subgradient-based recursive (R-) Lasso scheme was introduced first to obviate the growing (with time) complexity and memory requirements of batch Lasso alternatives. Simulations illustrated that R-Lasso outperforms least-squares and recursive least-squares (RLS) schemes that do not account for sparsity when estimating time-invariant and slow-varying signals of interest. Performance analysis revealed that R-Lasso estimates cannot simultaneously recover the signal support and the nonzero signal amplitudes consistently. This shortcoming prompted the development of an RLS-weighted (RW-) Lasso modification, which for proper selection of design parameters can be rendered provably support- *and* amplitude-consistent for time-invariant sparse signals. For time-varying sparse signals neither R-Lasso nor RW-Lasso can provide such double consistency guarantees, but simulated test cases demonstrated that the latter outperforms the former.

7. REFERENCES

[1] D. Angelosante and G. B. Giannakis, “RLS-weighted Lasso for adaptive estimation of sparse signals,” *IEEE Trans. on Signal Proc.*, 2008 (submitted).

[2] J.-A. Bazerque and G. B. Giannakis, “Distributed spectrum sensing for cognitive radios by exploiting sparsity,” *Proc. of 42nd Asilomar Conf.*, Pacific Grove, CA, Oct. 26-29, 2008.

[3] D. Bertsekas, *Non Linear Programming*, Second Edition, Athena Scientific, 2003.

[4] E. Candès and T. Tao, “Near optimal signal recovery from random projections: Universal encoding strategies?,” *IEEE Trans. on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[5] S. S. Chen, D. L. Donoho, M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[6] J. Fan and R. Li, “Variable selection via nonconcave penalized likelihood and its oracle properties,” *Journal Amer. Statist. Assoc.*, vol. 96, pp. 1348–1360, 2001.

[7] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice Hall, 1993.

[8] D. M. Malioutov, S. Sanghavi and A. S. Willsky, “Compressed sensing with sequential observations,” *Proc. of ICASSP*, Las Vegas, NV, April 2008.

[9] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *J. Royal. Statist. Soc.*, vol. 58, no. 1, pp. 267–288, 1996.

[10] P. Zhao and B. Yu, “On model selection consistency of Lasso,” *Journ. of Mach. Learning Res.*, vol. 7, pp. 2541–2563, 2006.

[11] H. Zou, “The adaptive Lasso and its oracle properties,” *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1418–1429, 2006.

[12] H. Zou and R. Li, “One-step sparse estimates in nonconcave penalized likelihood models,” *The Annals of Statistics*, vol. 36, no. 4, pp. 1509–1533, 2008.