GENERALIZED THRESHOLDING SPARSITY-AWARE ALGORITHM FOR LOW COMPLEXITY ONLINE LEARNING

Yannis Kopsinis[1], *Konstantinos Slavakis*[2], *Sergios Theodoridis*[3], *Steve McLaughlin*[4]

[1]University of Granada
Spain, kopsinis@ieee.org

[2]University of Peloponnese
Greece, slavakis@uop.gr

[3]The University of Edinburgh
UK, S.McLaughlin@hw.ac.uk

[4]University of Athens
Greece, stheodor@di.uoa.gr

ABSTRACT

In this paper, a novel scheme for online, sparsity-aware learning is presented. A new theory is developed that allows for the incorporation, in a unifying way, of different thresholding rules to promote sparsity, that may even be of a nonconvex nature. The complexity of the algorithm exhibits a linear dependence on the number of free parameters.

Index Terms— Adaptive filtering, sparsity, thresholding operators, signal recovery.

1. INTRODUCTION

Sparsity-aware learning has been a topic at the forefront of research over the last ten years or so [1]. Considerable research effort has been invested in developing efficient schemes for the recovery of sparse signal/parameter vectors. However, most of these efforts have been and continue to be focussed on batch processing where all measurements, training data, are available prior to the estimation task. It is only very recently that online, time-adaptive algorithms have been developed [2–4]. Moreover, the major evolution of these is along a greedy-like philosophy and the ℓ_1-norm regularization of a Least Squares (LS) regression term (LASSO-types). Equivalently, most of the algorithms build around the hard and soft thresholding operators, shown in Fig. 1a, in order to impose sparsity.

It is by now well established, in particular in the statistics community, that selecting the thresholding operator is a critical step, that can significantly affect the variance and the sensitivity of the resulting estimate [5–7]. To this end, a number of alternative to hard and soft thresholding rules have been proposed, in an effort to bypass their drawbacks. In general, this is achieved by modifying the regularization term, which can also be a nonconvex function.

In [8], an efficient sparsity-aware online algorithm was developed, in the context of the very recent advances of set theoretic estimation philosophy [9]. Sparsity was induced by constraining the solution to lie within the weighted ℓ_1 ball, which turned out to be equivalent to a soft thresholding rule.

The main goal of this paper is to develop a new online, sparsity-aware scheme that can employ different thresholding rules, in a *unifying* way. To this end, and since some of the thresholding rules are associated with nonconvex functions, the currently available theory of [9], must first be generalized. The current theory is built around projections onto convex sets. For the needs of the current paper, the theory has to include more general mapping operators, that allow for treatment of constraints associated with nonconvex sets. In particular, we will constraint our solution to lie in a union of subspaces, which is a nonconvex region. In the sequel, the possibilities of the new scheme will be exploited to derive efficient computational schemes, by exploiting different thresholding rules.

This work was supported in part by the Ramón y Cajal program.

2. SYSTEM MODEL AND PROBLEM STATEMENT

We will denote the set of all integers, nonnegative integers, positive integers, and real numbers by $\mathbb{Z}$, $\mathbb{N}$, $\mathbb{N}_*$, and $\mathbb{R}$, respectively. Given two integers $j_1, j_2 \in \mathbb{Z}$, such that $j_1 \leq j_2$, let $\overline{j_1, j_2} := \{j_1, j_1 + 1, \ldots, j_2\}$.

The stage of discussion will be the Euclidean space $\mathbb{R}^L$, where $L \in \mathbb{N}_*$. Given any couple of vectors $\boldsymbol{a}_1, \boldsymbol{a}_2 \in \mathbb{R}^L$, the inner product in $\mathbb{R}^L$ is defined as the classical vector-dot product $\langle \boldsymbol{a}_1, \boldsymbol{a}_1 \rangle := \boldsymbol{a}_1^t \boldsymbol{a}_2$, where the superscript t stands for vector/matrix transposition. The induced norm will be denoted by $\|\cdot\|$.

Our task is to estimate the signal $\boldsymbol{a}_* \in \mathbb{R}^L$, based on measurements that are sequentially generated by the linear regression model:

$$y_n = \boldsymbol{u}_n^t \boldsymbol{a}_* + v_n, \quad \forall n \in \mathbb{N}, \tag{1}$$

where the model outputs (observations) $(y_n)_{n\in\mathbb{N}} \subset \mathbb{R}$, and the model input vectors $(\boldsymbol{u}_n)_{n\in\mathbb{N}} \subset \mathbb{R}^L$ comprise the measurement pairs $(\boldsymbol{u}_n, y_n)_{n\in\mathbb{N}}$, and $(v_n)_{n\in\mathbb{N}}$ is the noise process. The unknown signal $\boldsymbol{a}_*$ is "sensed" by a sequence of inner products, with appropriately selected "sensing" vectors $\boldsymbol{u}_n$.

In this study, the signal $\boldsymbol{a}_*$ is assumed to be sparse, i.e., most of its components are zero. If we define $\|\boldsymbol{a}_*\|_0$ to stand for the number of nonzero components of $\boldsymbol{a}_*$, then the assumption that $\boldsymbol{a}_*$ is sparse can be equivalently given by $K := \|\boldsymbol{a}_*\|_0 \ll L$. Hereafter, such signals will be referred to as K-sparse.

3. THE SET THEORETIC ESTIMATION APPROACH TO ONLINE LEARNING

The mainstream of sparsity-aware online methods follows the classical path of adaptive filtering [10], where a quadratic objective function is used to quantify the designer's perception of loss. Such a convex differentiable function is then regularized, by a sparsity promoting term; the latter usually revolves around the ℓ_1 norm, and a minimizer of the resulting optimization task is sought either by the RLS or the LMS rationales, e.g., [2, 3]. Very recently, a novel online method for the recovery of sparse signals, referred to as the Adaptive Projection-based Algorithm, using Weighted ℓ_1-balls (APWL1) to promote sparsity, was developed in [8].

The philosophy behind [8] departs from the classical approach, and searches for a *set* of solutions which are in *agreement* with the available measurements as well as the a-priori knowledge. More specifically, at each time instance, $n \in \mathbb{N}$, the training data pair $(\boldsymbol{u_n}, y_n)$ is used to define a closed convex subset of $\mathbb{R}^L$, which is considered to be the region where the unknown $\boldsymbol{a}_*$ lies with high probability [9].

A plethora of alternatives exist on how to "construct" such convex regions. A popular choice takes the form of a *hyperslab* around $(\boldsymbol{u_n}, y_n)$, which is defined as:

$$S_n[\epsilon] := \left\{ \boldsymbol{a} \in \mathbb{R}^L : \ |\boldsymbol{u}_n^t \boldsymbol{a} - y_n| \leq \epsilon \right\}, \quad \forall n \in \mathbb{N}, \tag{2}$$

for some user-defined tolerance $\epsilon \geq 0$, and for $\boldsymbol{u}_n \neq \boldsymbol{0}$. The parameter ϵ determines, the width of the hyperslabs, and essentially models the noise effect, as well as various other uncertainties, like measurement inaccuracies, calibration errors, etc. Any point that lies in the hyperslab is considered to be in agreement with the corresponding measurement pair, at the specific time instance. For example, if the noise were bounded, then for any $(\boldsymbol{u_n}, y_n)$ and a careful choice of ϵ, it would be certain that the unknown solution would lie within $S_n[\epsilon]$.

Our ultimate goal, given the sequence of training pairs $(\boldsymbol{u}_n, y_n)_{n\in\mathbb{N}}$, is to search for a point $\hat{\boldsymbol{a}}_* \in \mathbb{R}^L$ that lies in the intersection of the hyperslabs $(S_n[\epsilon])_{n\in\mathbb{N}}$, which are defined by the training points. This is achieved via a sequence of projections onto these hypeslabs.

A notable characteristic of the adaptive set theoretic rationale is the simplicity with which convex constraints, (which encode a-priori knowledge other than the training sequence), can be accommodated. The generic algorithmic step comprises a single recursion; starting from an arbitrary $\boldsymbol{a}_0 \in \mathbb{R}^L$,

$$\boldsymbol{a}_{n+1} := T_n \left(\boldsymbol{a}_n + \mu_n \left(\sum_{i=n-q+1}^{n} \omega_i^{(n)} P_{S_i[\epsilon]}(\boldsymbol{a}_n) - \boldsymbol{a}_n \right) \right), \tag{3}$$

where the extrapolation parameter $\mu_n \in (0, 2\mathcal{M}_n)$, with

$$\mathcal{M}_n := \begin{cases} \dfrac{\sum_{n-q+1}^{n} \omega_i^{(n)} \left\| P_{S_i[\epsilon]}(\boldsymbol{a}_n) - \boldsymbol{a}_n \right\|^2}{\left\| \sum_{n-q+1}^{n} \omega_i^{(n)} P_{S_i[\epsilon]}(\boldsymbol{a}_n) - \boldsymbol{a}_n \right\|^2}, \\ \qquad\qquad \text{if } \sum_{n-q+1}^{n} \omega_i^{(n)} P_{S_i[\epsilon]}(\boldsymbol{a}_n) \neq \boldsymbol{a}_n, \\ 1, \qquad \text{otherwise}, \end{cases} \tag{4}$$

$P_{S_i[\epsilon]}$ stands for the (metric) projection mapping onto the hyperslab $S_i[\epsilon]$, and $\{\omega_i^{(n)}\}_{i=n-q+1}^{n} \subset (0, 1]$ is a set of weights, such that $\sum_{i=n-q+1}^{n} \omega_i^{(n)} = 1$. The sequence of mappings $(T_n : \mathbb{R}^L \to \mathbb{R}^L)_{n\in\mathbb{N}}$ encapsulates all the constraints on the signal under study. Very recent advances, on the study of the recursions (3) and (4) [11], have revealed that such operators belong to the very general class of quasi-nonexpansive mappings, with a remarkable flexibility on describing a large variety of convex constraints. The present study is the first step beyond [11], to the case where a-priori information takes the shape of *non-convex* sets.

In [8], the place of $(T_n)_{n\in\mathbb{N}}$ is taken by the projection mappings onto a sequence of weighted ℓ_1-balls. This scheme is equivalent to coordinate-wise soft thresholding operations. In this paper, we go one step further by considering sparsity-promoting generalized thresholding operators. It should be emphasized, that the proposed thresholding rules are neither restricted to be continues nor to satisfy the "hard to deal without it" convexity requirement. Our generalized thresholding (GT) operator $T_{\text{GT}}^{(K)}$, which will replace T_n in (3), is described next.

4. THE GENERALIZED THRESHOLDING OPERATOR

Given a $K \in \overline{1, L}$, the Generalized Thresholding (GT) operator $T_{\text{GT}}^{(K)} : \mathbb{R}^L \to \mathbb{R}^L$ is defined as follows; for any $\boldsymbol{x} \in \mathbb{R}^L$, the output $\boldsymbol{z} := T_{\text{GT}}^{(K)}(\boldsymbol{x})$, is obtained coordinate-wise as follows:

$$\forall l \in \overline{1, L}, \quad z_l := \begin{cases} x_l, & l \in J_{\boldsymbol{x}}^{(K)}, \\ \text{shr}(x_l), & l \notin J_{\boldsymbol{x}}^{(K)}, \end{cases} \tag{5}$$

where $J_{\boldsymbol{x}}^{(K)}$ contains all those positions (indices), which correspond to the K largest, in absolute value, components of the vector $\boldsymbol{x}$, and shr denotes a user-defined shrinkage function. In simple words, GT acts as follows: given the input vector $\boldsymbol{x} \in \mathbb{R}^L$, identify, first, its K largest, in magnitude, components, while apply to the rest of them the shrinkage function shr. Define $\xi_{\boldsymbol{x}}^{(K)} := \min\{|x_l| : l \in J_{\boldsymbol{x}}^{(K)}\}$. That is, $\xi_{\boldsymbol{x}}^{(K)}$ stands for one of the K-th largest components, in magnitude, of the vector $\boldsymbol{x}$. Clearly, $\forall l \notin J_{\boldsymbol{x}}^{(K)}$, $|x_l| \leq \xi_{\boldsymbol{x}}^{(K)}$. Then, $\text{shr} : \mathbb{R} \to \mathbb{R}$ is required to satisfy the following properties:

1. $\tau \, \text{shr}(\tau) \geq 0, \forall \tau \in \mathbb{R}$.
2. Given $\epsilon > 0$, there exists a $\delta > 0$ such that $\forall \tau$, which satisfy $\epsilon \leq |\tau| \leq \xi_{\boldsymbol{x}}^{(K)}$, we have $|\,\text{shr}(\tau)| \leq |\tau| - \delta$. That is, shr acts as a *strict* shrinkage operator over the intervals which do not include 0. The upper bound $\xi_{\boldsymbol{x}}^{(K)}$ of the interval is not restrictive at all; recall that since, by definition, shr applies to all but the K largest, in magnitude, components of the vector $\boldsymbol{x}$, it is natural for the arguments of shr to be less than or equal to $\xi_{\boldsymbol{x}}^{(K)}$.

Any arbitrary function, which is inline with the properties above, can be used for shr. Such an example is shown in Fig. 1b. Moreover, shr can be substituted with *any* thresholding operator which solves the univariate penalized least squares problem [5,7] with convex or even non-convex penalty functions. Therefore, a number of well known operators can be directly integrated into the proposed framework. Examples of such operators are those related to the ℓ_γ penalty, for $\gamma \in [0, 1]$, the log-, the SCAD, the MC+, and the transformed ℓ_1 penalties [5, 7]. In Fig. 1c, two examples of the GT are shown, where as shrinkage function shr the 2-degree garrote (solid line) and the thresholding function associated with the bridge, $\ell_{0.5}$ penalty (dashed line) have been chosen.

It can be verified (omitted due to lack of space) that the sparsity-cognizant $T_{\text{GT}}^{(K)}$ associates to a non-convex constraint set. More specifically, it is intimately connected to the union $\bigcup_J M_J$, where J is any selection of K positions in an L-dimensional vector, and $M_J := \{\boldsymbol{x} \in \mathbb{R}^L : x_l = 0, \forall l \notin J\}$. It can be readily verified that each M_J is a linear subspace of $\mathbb{R}^L$, and $\bigcup_J M_J$ is thus non-convex. Despite this fact, it can be shown (omitted due to lack of space), that under some mild assumptions, the algorithmic scheme of (3), (4), with T_n substituted by $T_{\text{GT}}^{(K)}$, leads to a sequence of estimates $(\boldsymbol{a}_n)_{n\in\mathbb{N}}$ whose set of cluster points is nonempty, each one of them is guaranteed to be, at most, K-sparse, and located arbitrarily close to an intersection of an infinite number of hyperslabs $S_n[\epsilon]$.

The developments regarding the proposed GT operator are of high interest. It is the first time that the specific algorithmic family is rendered capable of incorporating non-convex constraints. In fact, to the best of our knowledge, there is not any adaptive algorithm, of linear computational complexity, capable of dealing with such constraints. Moreover, the flexibility of the GT can lead to efficient sparsity inducing thresholding rules targeted to high performance implementations of reduced complexity.

A thorough study of the forms that GT can take as well as their implications in practice is beyond the scope of this paper. Here, we focus on sparsity-inducing thresholding operators, which lead to reduced computational complexity, compared to the previously used projections onto weighted ℓ_1 balls. More specifically, the sparsity promotion via projections onto weighted ℓ_1 balls [8] has the following drawbacks: a) it does not lead to estimates with a fixed and predefined sparsity level at each iteration, b) it requires $\mathcal{O}(L)$ multiplications and divisions, and c) it requires a full sorting of the unknown vector values, which takes $\mathcal{O}(L \log_2 L)$ sorting operators.

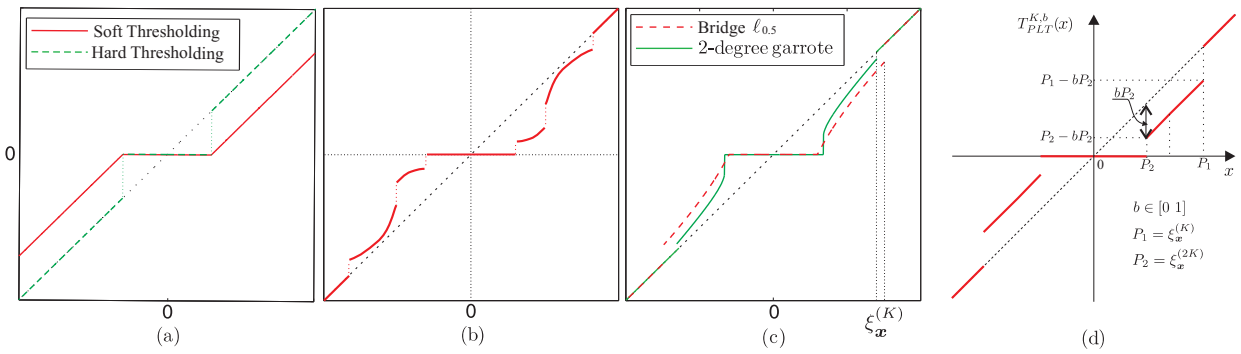

Fig. 1. Several thresholding functions.

5. LOW COMPLEXITY SPARSITY INDUCING OPERATORS

The less complex thesholding rule is inevitably the hard thresholding (HT) one. In the general context of our GT formulation, the HT rule becomes: Having an estimate K of the actual sparsity level S, HT sets all but the largest (in magnitude) K components of $\boldsymbol{x}$ to zero. In the cases where the K larger components might not be uniquely defined, then the smallest possible indices are chosen. The respective computational complexity comprises the detection of the Kth order statistic of $\boldsymbol{x}$, which can be performed in linear time, $\mathcal{O}(L)$. Moreover, HT leads to estimates per iteration with fixed sparsity level, which is equal to the predetermined value K.

A major drawback of HT is that it sets to exactly zero all the $L-K$ smaller components. However, such a "strict" strategy may push to zero coefficients which, actually, belong to the support, especially at the early stages of the algorithm, prior to convergence, where the obtained estimates may not be good. A more "gentle" treatment, would be, instead of zeroing these $L-K$ values, to shrink their values to some degree. Several such strategies have been proposed in the literature, with the celebrated SCAD penalty being one of them. Here, we propose and study a thresholding rule, which is simpler than the SCAD, referred to as piecewise linear thresholing (PLT) rule, $\boldsymbol{z} = T_{PLT}^{K}(\boldsymbol{x}_n)$, which operates coordinate-wise as follows:

$$z_i = \begin{cases} 0, & \text{if } |x_i| \leq P_2 \\ \text{sgn}(x_i)(|x_i| - bP_2), & \text{if } P_2 < |x_i| \leq P_1 \\ x_i, & \text{if } |x_i| > P1 \end{cases}, \qquad (6)$$

where, $P_1 = \xi_{(x)}^{(K)}$, $P_2 = \xi_{(x)}^{(2K)}$, and b is a free parameter taking values in $[0\ 1]$.

Schematically, the PLT operation is shown in Fig. 1d. It can be observed that it shares common attributes with both hard and soft thresholding. The major advantage of PLT, is that it can be essentially considered as a multiplication free operator (it needs 1 multiplication only). Moreover, similarly to HT, PLT leads to estimates of fixed sparsity level, equal to $2K$.

With respect to computational complexity, the basic recursive scheme of (3), at each iteration, requires approximately $qK_n + qL\beta$ multiplications, where K_n is the sparsity level of the current estimate and the value of β depends on the specific configuration. Usually in practice, all $\omega_i^{(n)}$ take the common value $1/q$, and also $\|u\| = 1$. In this case, $\beta = 1$. For the APWL1 case, where the sparsity level is not fixed in each iteration, then the worse case scenario, where $K_n = L$, should be considered. On the contrary, for thresholding rules such as HT and PLT, K_n equals to K and $2K$ respectively. In other words, for highly sparse signals, the proposed algorithms roughly halves the number of required multiplications. On top of that, HT and PLT are essentially multiplication and division-free operators.

6. SIMULATION EXAMPLES

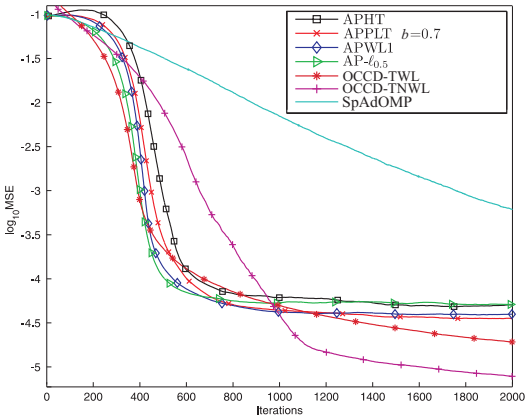

Fig. 2. Performance evaluation in time-constant conditions.

Fig. 2 shows comparison results of the proposed algorithms for different choices of the sparsity inducing operator $T_{GT}^{(K)}$, where $L := 1024$ and $S := 100$, with the noise variance set equal to $\sigma^2 = 0.1$. All the adaptive projection-based algorithms use $q = 390$, which was the largest q, after extensive experimentation, that gave the best performance. Moreover, the extrapolation parameter, μ_n is set equal to $\mathcal{M}_n$, [8], each $\omega_j^{(n)}$ is defined to be $1/q$, and the hyperslabs parameter $\epsilon := 1.3 \times \sigma$. In all the cases, the measurement vectors $\boldsymbol{u}_n$ are normally distributed.

The performance of the proposed low complexity adaptive projection-based algorithms, using HT and PLT (referred to as and APHT and APPLT respectively), is shown with curves indicated with squares and x-crosses respectively. Parameter b was set equal to 0.7 since it gave the faster convergence speed. Clearly, PLT leads to both faster convergence and lower error floor approaching the performance succeeded by the much complex APWL1. In order to show the potential for further performance improvements based on GT, the results when using the bridge, $\ell_{0.5}$ thresholding are also depicted (curve denoted by triangles). It is observed that

this latter configuration led to somewhat faster convergence compared to APWL1. The proposed methods are compared against state-of-the-art sparsity aware online algorithms, such as the Online Cyclic Coordinate Descent - Time Weighted Lasso (OCCD-TWL) (curve marked with asterisks) and Time and Norm Weighted LASSO (OCCD-TNWL)(curve marked with crosses) [2]. These online algorithms are of great interest since they succeed in attaining performance similar to LASSO using closed form adaptation equations. Concerning computational complexity, OCCD-TWL needs more than $2L^2$ multiplications, whereas OCCD-TNWL roughly scores the double of that complexity. Moreover, the performance of SpAdOMP [3], which is considered one of the best low complexity sparsity aware adaptive algorithms, is shown with the unmarked solid line. Clearly, SpAdOMP needs many more iterations in order to converge, so it can not compete with the rest of the algorithms, which are computationally more demanding. It should be stressed that the proposed algorithmic scheme is inherently capable of operating with complexities similar to SpAdOMP, by using low q values. We have observed, that our algorithm can be especially benefited with proper choices of GT's in this low complexity operational setup. However, in this paper, we focus on large q values, in order to study the proposed algorithms when they achieve their full performance.

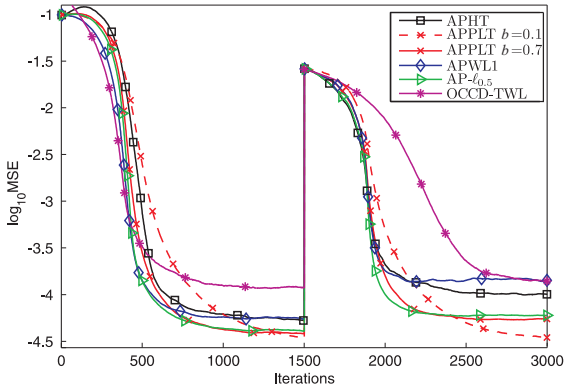

Fig. 3. Performance evaluation in time-varying conditions.

Fig. 3 shows the ability of the tested algorithms to track an abrupt change, which is realized after 1500 observations. Particularly, in the first half, the signal under consideration has the characteristics of that discussed in Fig. 2. However, at the mid-time point of 1500, 10 randomly selected components, change their values from 0 to a randomly selected nonzero one. Thus, the signal after 1500 has a sparsity level of 110. In the specific example, K is set equal to 100, so in the second half, the AP methods operate with an underestimate of the true sparsity level. It can be concluded that AP-$\ell_{0.5}$ and APHT appear to be more sensitive compared to the rest of the proposed methods. Moreover, the APPLT, with $b = 0.1$, reaches lower error floors, albeit at a slower convergence speed. In general, it can be seen that the choice of the sparsity inducing operator is a factor with high potential for the development of efficient variants of the adopted algorithmic family. For comparison, the OCCD-TWL with forgetting factor equal to 0.996 is depicted, with the associated curve marked with asterisks.

7. CONCLUSIONS

A novel online sparsity aware scheme of linear complexity has been presented, that can deal in a unifying way different thresholding rules, including nonconvex ones.

8. REFERENCES

[1] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[2] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: Where RLS meets the ℓ_1-norm," *IEEE Trans. Signal Proc.*, vol. 58, no. 7, pp. 3436–3447, July 2010.

[3] G. Mileounis, B. Babadi, N. Kalouptsidis, and V. Tarokh, "An adaptive greedy algorithm with application to nonlinear communications," *IEEE Trans. Signal Proc.*, vol. 58, no. 6, pp. 2998–3007, June 2010.

[4] Y. Murakami, M. Yamagishi, M. Yukawa, and I. Yamada, "A sparse adaptive filtering using time-varying soft-thresholding techniques," in *Proceedings of the IEEE ICASSP*, Dallas: USA, Mar. 2010, pp. 3734–3737.

[5] A. Antoniadis and J. Fan, "Regularization of wavelet approximations," *Journal of the American Statistical Association*, vol. 96, pp. 939–967, 2001.

[6] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of The American Statistical Association*, vol. 96, pp. 1348–1360, 2001.

[7] R. Mazumder, J. H. Friedman, and T. Hastie, "Sparsenet: Coordinate descent with nonconvex penalties," *Journal of the American Statistical Association (JASA)*, 2011, to appear.

[8] Y. Kopsinis, K. Slavakis, and S. Theodoridis, "Online sparse system identification and signal reconstruction using projections onto weighted ℓ_1 balls," *IEEE Trans. Signal Proc.*, vol. 59, no. 3, pp. 905–930, Mar. 2011.

[9] S. Theodoridis, K. Slavakis, and I. Yamada, "Adaptive learning in a world of projections," *Signal Processing Magazine, IEEE*, vol. 28, no. 1, pp. 97–123, Jan. 2011.

[10] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New Jersey: John Wiley & Sons, 2003.

[11] K. Slavakis and I. Yamada, "The adaptive projected subgradient method constrained by families of quasi-nonexpansive mappings and its application to online learning," 2011, conditionally accepted for publication in the SIAM J. Optimization. [Online]. Available: http://arxiv.org/abs/1008.5231