ONLINE COORDINATE DESCENT FOR ADAPTIVE ESTIMATION OF SPARSE SIGNALS

Daniele Angelosante, Juan Andres Bazerque, Georgios B. Giannakis

University of Minnesota, Dept. of ECE, Minneapolis, MN 55455, USA

ABSTRACT

Two low-complexity sparsity-aware recursive schemes are developed for real-time adaptive signal processing. Both rely on a novel online coordinate descent algorithm which minimizes a time-weighted least-squares cost penalized with the scaled ℓ_1 norm of the unknown parameters. In addition to computational savings offered when processing time-invariant sparse parameter vectors, both schemes can be used for tracking slowly varying sparse signals. Analysis and preliminary simulations confirm that when the true signal is sparse the proposed estimators converge to a time-weighted least-absolute shrinkage and selection operator, and both outperform sparsity-agnostic recursive least-squares alternatives.

Index Terms— Lasso, Basis Pursuit, Compressive Sensing, Coordinate Descent, Recursive Least-Squares

1. INTRODUCTION

The least-absolute shrinkage and selection operator (Lasso), basis pursuit, and the most recent approaches of compressive sensing offer a suite of tools to deal with sparse signals that emerge in various applications; see e.g., [1, 2]. Whether for regression, compression, sampling or reconstruction, most of these tools enable sparsity-aware batch processing of the available (possibly noisy) observations. On the other hand, it is well appreciated that complexity considerations and the desire to cope with (at least slowly) varying signals motivate adaptive processing of sequentially acquired observations. Sequential recovery of noise-free sparse signals is considered in [3], and sparsity-aware least mean-square (LMS) based adaptive estimation is explored in [4].

The starting point here is the recursive least-squares (RLS) algorithm, which minimizes a sequence of LS cost functions formed by sequentially acquired observations; see e.g., [5]. To effect sparsity, the RLS cost is penalized with the scaled ℓ_1 norm of the unknown vector as in [6], where a time-weighted (TW) recursive Lasso estimator is introduced. The main limitation of [6] is either prohibitive complexity when a non-linear optimization (convex) solver is invoked per datum, or, slow convergence when a sub-gradient iteration is adopted for real-time operation. This paper is prompted by the recent *offline* application of the cyclic coordinate descent (CCD, a.k.a. Gauss-Seidel) algorithm adopted to solve *batch* Lasso problems [7, 8]. Instead, the present contribution develops two novel *online* coordinate descent (OCD) algorithms offering *adaptive* TW-Lasso estimates with desirable complexity versus convergence tradeoffs. Convergence analysis and simulations demonstrate the attractive performance of the proposed algorithms in estimating sparse time-invariant parameter vectors and tracking of slowly-varying sparse signals.

The problem is stated in Section 2. The two online algorithms are developed in Section 3, where almost-sure convergence is also asserted. Section 4 contains simulations to test performance of the proposed schemes, while concluding remarks are given in Section 5.

Notation: Vectors (matrices) are denoted with lower-case (upper case) boldface letters; and ^{T} stands for transposition. The pth entry of $\mathbf{x}$ is denoted as $x(p)$, and the (p,q)th entry of matrix $\mathbf{R}$ as $R(p,q)$. Function $\mathcal{N}(\mu,\sigma^2)$ stands for the Gaussian density function with mean μ and variance σ^2.

2. PRELIMINARIES AND PROBLEM STATEMENT

Consider a vector $\mathbf{x}_o \in \mathbb{R}^P$ that is sparse in the sense that only a few of its entries $x_o(p)$, $p = 1,\ldots,P$, are nonzero. Let $\mathcal{S}_{\mathbf{x}_o} := \{p : x_o(p) \neq 0\}$ denote its support, $P_1 := |\mathcal{S}_{\mathbf{x}_o}|$ the number of its non-zero entries, and $P_0 := P - P_1$. Sparsity amounts to having $P_1 \ll P_0$. Suppose that such a sparse vector is to be estimated sequentially in time from scalar observations obeying the linear regression model

$$y_n := \mathbf{h}_n^T \mathbf{x}_n + v_n, \quad n = 1,\ldots,N \tag{1}$$

where $\mathbf{h}_n \in \mathbb{R}^P$ denotes the known regression vector at time n, and the additive noise v_n is assumed white with zero mean and variance σ^2.

Based on y_n, the goal is to develop online estimators of the sparse $\mathbf{x}_o$ when the latter is time-invariant or slowly time varying. If $\mathbf{x}_o$ were non-sparse, the RLS algorithm would have been the natural choice for such a task. At computational

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.

cost $\mathcal{O}(P^2)$ per datum, the RLS estimate is given by [5]

$$\widehat{\mathbf{x}}_N^{\text{RLS}} := \arg\min_{\mathbf{x}\in\mathbb{R}^P} \sum_{n=1}^{N} \beta_{N,n}(y_n - \mathbf{h}_n^T\mathbf{x})^2, \quad N = 1, 2, \ldots \tag{2}$$

where $\beta_{N,n}$ captures either one of the following two choices for data windowing:

(c1) Infinite window with $\beta_{N,n} = 1$. If $\mathbf{x}_o$ is time invariant, then RLS is (at least for large enough N) equivalent to the batch LS, which incurs complexity $\mathcal{O}(P^3)$.

(c2) Exponentially decaying window with $\beta_{N,n} = \beta^{N-n}$, where thanks to the forgetting factor $\beta \in (0, 1)$ the RLS estimate can track slowly varying signals.

However, RLS does not exploit the sparsity present in $\mathbf{x}_o$. Motivated by the well documented merits of the ℓ_1 norm for sparse regression, a sequence of ℓ_1 penalized estimation problems is considered, to obtain time-weighted Lasso (TW-Lasso) estimates as (see also [6])

$$\widehat{\mathbf{x}}_N^{\text{TWL}} := \arg\min_{\mathbf{x}\in\mathbb{R}^P} \left[\frac{1}{2}\sum_{n=1}^{N} \beta_{N,n}(y_n - \mathbf{h}_n^T\mathbf{x})^2 + \lambda_N\|\mathbf{x}\|_1\right] \tag{3}$$

where $\|\cdot\|_1$ denotes the ℓ_1 norm, and the subscript of λ highlights the possible adaptation of the penalty term with time.

The vector iterates developed in the next section to estimate $\mathbf{x}_o$ online must be sparse, they should approximate the sequence $\{\widehat{\mathbf{x}}_N^{\text{TWL}}\}$ of solutions to (3), and have to converge to $\mathbf{x}_o$ at least when the unknown vector is time-invariant.

3. ONLINE COORDINATE DESCENT TW-LASSO

Recall the minimization problem in (3), namely $\min_{\mathbf{x}} J_N(\mathbf{x})$, where $J_N(\mathbf{x}) := \frac{1}{2}\sum_{n=1}^{N} \beta_{N,n}(y_n - \mathbf{h}_n^T\mathbf{x})^2 + \lambda_N\|\mathbf{x}\|_1$. One approach to finding the solution $\widehat{\mathbf{x}}_N^{\text{TWL}}$ is to run a CCD algorithm, which in its simplest form entails cyclic iterative minimization of $J_N(\mathbf{x})$ with respect to (w.r.t.) one coordinate per iteration. Let $\mathbf{x}_N^{(i-1)}$ denote the solution at iteration $i-1$. The pth variable at the ith iteration is updated as

$$x_N^{(i)}(p) := \arg\min_{x} \; J_N([x_N^{(i)}(1), \ldots, x_N^{(i)}(p-1), x, x_N^{(i-1)}(p+1), \ldots, x_N^{(i-1)}(P)]). \tag{4}$$

Albeit convex, the cost $J_N(\mathbf{x})$ is non-differentiable. Nonetheless, convergence of the CCD iterates for Lasso-type problems follows readily using the results of [9]. In addition to affording effective initialization (with the all-zero vector), an attractive feature of CCD Lasso solvers is that each coordinate-wise minimizer is available in closed form.

The online coordinate descent (OCD) algorithm introduced next can be viewed as a challenging adaptive counterpart of CCD Lasso, where a new datum is incorporated at each iteration. The challenge arises because the cost function changes with N. For notational convenience, express the time index as $N = kP + p$, where $p \in \{1, \ldots, P\}$ corresponds to the entry (coordinate) of $\mathbf{x}$ to be updated at time N, and $k = \lceil \frac{N}{P} \rceil - 1$ indicates the number of cycles; that is, how many times the pth coordinate has been updated. Let $\widehat{\mathbf{x}}_{N-1}^{\text{OCD}}$ denote the solution of the OCD algorithm at time $N-1$. With $J_N(\mathbf{x})$ as in (3), let $\widehat{x}_N^{\text{OCD}}(q) = \widehat{x}_{N-1}^{\text{OCD}}(q)$ for $q \neq p$, and optimize only the pth component in a coordinate descent fashion as

$$\widehat{x}_N^{\text{OCD}}(p) := \arg\min_{x} J_N([\widehat{x}_{N-1}^{\text{OCD}}(1), \ldots, \widehat{x}_{N-1}^{\text{OCD}}(p-1), x, \widehat{x}_{N-1}^{\text{OCD}}(p+1), \ldots, \widehat{x}_{N-1}^{\text{OCD}}(P)]) \; . \tag{5}$$

In the cyclic update (5), the coordinates $\{\widehat{x}_{N-1}^{\text{OCD}}(q),\ q < p\}$ are updated $k+1$ times, while entries $\{\widehat{x}_{N-1}^{\text{OCD}}(q),\ q > p\}$ are updated k times [cf. (4)]. To fully specify the OCD TW-Lasso, define also

$$\mathbf{r}_N := \sum_{n=1}^{N} \beta_{N,n} y_n \mathbf{h}_n, \qquad \mathbf{R}_N := \sum_{n=1}^{N} \beta_{N,n} \mathbf{h}_n \mathbf{h}_n^T. \tag{6}$$

After isolating only terms which depend on the coordinate x that is currently optimized, recursion (5) can be rewritten as

$$\widehat{x}_N^{\text{OCD}}(p) = \arg\min_{x} \frac{1}{2} R_N(p,p)x^2 - r_{N,p}x + \lambda_N|x| \tag{7}$$

$$r_{N,p} := r_N(p) - \sum_{q\neq p} R_N(p,q)\widehat{x}_{N-1}^{\text{OCD}}(q). \tag{8}$$

Being a scalar optimization problem, the Lasso estimate in (7) accepts a closed-form solution expressed as a soft-thresholding operation, namely

$$\widehat{x}_N^{\text{OCD}}(p) = \frac{\text{sgn}(r_{N,p})}{R_N(p,p)}\big[|r_{N,p}| - \lambda_N\big]_+ \tag{9}$$

where $[\cdot]_+$ denotes projection over the nonnegative reals. Equation (9) amounts to a soft-thresholding operation that sets to zero nonactive entries, thus facilitating convergence to sparse iterates.

The OCD Lasso scheme is tabulated as Algorithm 1. The

Algorithm 1 OCD TW-Lasso

Initialize $\widehat{\mathbf{x}}_0^{\text{OCD}} = \mathbf{0},\ p = 1, \ldots, P$
for $k = 0, 1, \ldots$ **do**
  for $p = 1, \ldots, P$ **do**
    S1. Acquire datum y_N, and regressor $\mathbf{h}_N$, $N = kP + p$.
    S2. Evaluate $\mathbf{r}_N$ and $\mathbf{R}_N$ as in (6).
    S3. Update $\widehat{x}_N^{\text{OCD}}(q) = \widehat{x}_{N-1}^{\text{OCD}}(q)$ for all $q \neq p$.
    S4. Compute $r_{N,p}$ via (8).
    S5. Update $\widehat{x}_N^{\text{OCD}}(p)$ via (9).
  end for
end for

number of algebraic operations involved in step S5 is $\mathcal{O}(P)$. Note also that $\mathbf{r}_N$ and $\mathbf{R}_N$ are updated online in S2, which leads to memory savings. Indeed, if e.g., (c2) is adopted, then $\mathbf{r}_N = \beta\mathbf{r}_{N-1} + y_N\mathbf{h}_N$ and $\mathbf{R}_N = \beta\mathbf{R}_{N-1} + \mathbf{h}_N\mathbf{h}_N^T$. On the other hand, updating $\mathbf{R}_N$ generally requires $\mathcal{O}(P^2)$ algebraic operations unless approximate OCD alternatives are sought. However, in transversal filtering and beamforming applications where sliding regressors are involved, i.e., when $\mathbf{h}_n := [h(n), h(n-1), \ldots, h(n-P-1)]^T$, matrix $\mathbf{R}_N$ can be updated exactly with linear complexity [10]. In these cases, OCD offers online estimation of sparse signals with very low overall complexity $\mathcal{O}(P)$.

To establish convergence of the OCD algorithm, the following conditions must be met:

(a1) $\lim_{N\to\infty} \frac{1}{N}\mathbf{R}_N = \mathbf{R}_\infty$ with probability (w.p.) 1

(a2) $\lim_{N\to\infty} \frac{1}{N}\mathbf{r}_N = \mathbf{r}_\infty$ w.p. 1.

Under these ergodicity (mixing) conditions that are typically satisfied in practice, it follows from the convergence of RLS that $\mathbf{R}_\infty^{-1}\mathbf{r}_\infty = \mathbf{x}_o$. Using the latter along with arguments similar to those involved in the stability of linear time-varying systems, we have proved the following result.[1]

Proposition 1 *Under (c1), (a1), and (a2), if* $\mathbf{x}_o$ *is time-invariant, and* $\lim_{N\to\infty} \frac{\lambda_N}{N} = 0$*, then* $\lim_{N\to\infty} \widehat{\mathbf{x}}_N^{OCD} = \mathbf{x}_o$ *w.p. 1.*

In words, Proposition 1 asserts that the OCD TW-Lasso estimates are (strongly) consistent.

3.1. Online Selective Coordinate Descent

The proposed OCD has low complexity but may exhibit slow convergence since each variable is updated every P observations. However, $P_1 \ll P_0$ due to sparsity, and most of the time OCD re-sets to zero entries of $\mathbf{x}_o$ that have been already set to zero. On the other hand, updating zero variables cannot be skipped a priori since nonzero entries may arise in time-varying scenarios. To address this dilemma, the idea in this section is to select which coordinate to update online in the same spirit used by [8] in the off-line CCD.

Let $d_{\mathbf{e}_p} J_N(\widehat{\mathbf{x}}_{N-1}^{\text{OSCD}})$ and $d_{\mathbf{e}_n} J_N(\widehat{\mathbf{x}}_{N-1}^{\text{OSCD}})$ denote the forward and backward directional derivatives w.r.t. $x(p)$ evaluated at $\widehat{\mathbf{x}}_{N-1}^{\text{OSCD}}$, the online selective coordinate descent (OSCD) estimate at time time $N-1$. Define also the vectors $\mathbf{d}^+, \mathbf{d}^- \in \mathbb{R}^P$ whose pth entries are $d_{\mathbf{e}_p} J_N(\widehat{\mathbf{x}}_{N-1}^{\text{OSCD}})$ and $d_{\mathbf{e}_n} J_N(\widehat{\mathbf{x}}_{N-1}^{\text{OSCD}})$, respectively. It holds that

$$\mathbf{d}^+ = \mathbf{R}_N\widehat{\mathbf{x}}_{N-1}^{\text{OSCD}} - \mathbf{r}_N + \lambda_N\mathbf{s}^+ \quad (10)$$

$$\mathbf{d}^- = \mathbf{r}_N - \mathbf{R}_N\widehat{\mathbf{x}}_{N-1}^{\text{OSCD}} + \lambda_N\mathbf{s}^- \quad (11)$$

with $\mathbf{s}^+, \mathbf{s}^- \in \mathbb{R}^P$, $s^+(p) = 1$ if $\widehat{\mathbf{x}}_{N-1}^{\text{OSCD}} \geq 0$ and $s^+(p) = -1$ otherwise; while $s^-(p) = 1$ if $\widehat{\mathbf{x}}_{N-1}^{\text{OSCD}} \leq 0$ and $s^+(p) = -1$ otherwise. The resultant OSCD scheme is summarized as Algorithm 2.

[1]Proof is omitted due to space limitations.

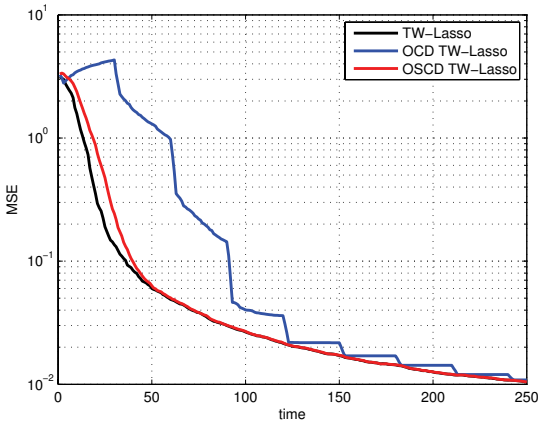

Fig. 1. MSE across time

Evaluation of each directional derivative entails multiplication of a $P \times P$ matrix by a $P \times 1$ vector, which requires $\mathcal{O}(P^2)$ operations. But since $\widehat{\mathbf{x}}_{N-1}^{\text{OSCD}}$ here is sparse, this product requires $\mathcal{O}(P_1 P)$ operations; and thus OSCD incurs complexity $\mathcal{O}(P)$. After evaluating (10) and (11), the coordinate with the most negative directional derivative, either forward or backward, is updated.

4. SIMULATIONS

The OCD and OSCD algorithms are tested here using two simulated examples.

Test case 1. Gaussian observations are generated according to (1) with a time-invariant $\mathbf{x}_o$, and parameters $P = 30$, $P_1 = 3$, $v_n \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = 10^{-1}$, and $\mathbf{h}_n \sim \mathcal{N}\left(\mathbf{0}_P, \frac{1}{P}\mathbf{I}_P\right)$. The penalty scale is set to $\lambda_N = \sqrt{2\sigma^2 N \log P}$, and (c1) windowing is utilized. The first three entries of $\mathbf{x}_o$ are chosen equal to unity, and all other entries are set to zero. Figure 1 depicts the mean-square error (MSE), i.e., $\text{E}[\|\widehat{\mathbf{x}}_n - \mathbf{x}_o\|^2]$, across time for the TW-Lasso in [6], along with the novel OCD and OSCD TW-Lasso estimates. As expected, OCD TW-Lasso exhibits slower convergence than the TW-Lasso (obtained by solving a sequence of second-order cone problems); but OSCD exhibits convergence similar to TW-Lasso at complexity comparable to OCD.

Test case 2. Gaussian observations are generated according to

Algorithm 2 OSCD TW-Lasso

Initialize $\widehat{\mathbf{x}}_0^{\text{OSCD}} = \mathbf{0}$, $p = 1, \ldots, P$.
for $N = 1, 2, \ldots$ **do**
S1. Acquire datum y_N, and regressor $\mathbf{h}_N$.
S2. Evaluate $\mathbf{d}^+$ and $\mathbf{d}_-$ as in (10) and (11).
S3. Select $p^* = \arg\min_p\{d^+(p), d^-(p)\}_{p=1}^P$.
S4. Update $\widehat{x}_N^{\text{OSCD}}(q) = \widehat{x}_{N-1}^{\text{OSCD}}(q)$ for all $q \neq p^*$.
S5. Compute r_{N,p^*} via (8).
S6. Update $\widehat{x}_N^{\text{OSCD}}(p^*)$ as in (9).
end for

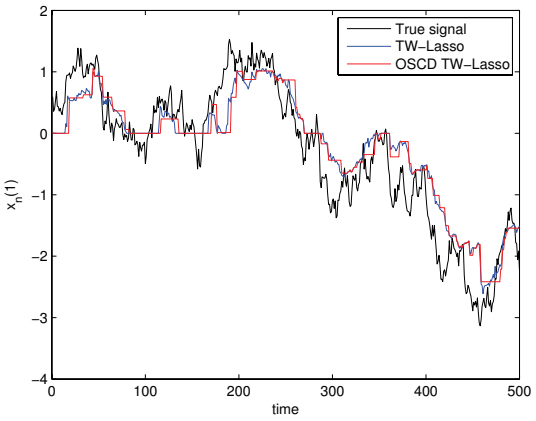

Fig. 2. Nonzero component estimates

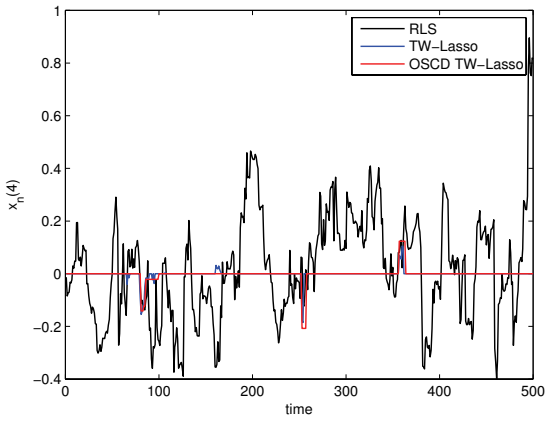

Fig. 3. Inactive component estimates

(1) with a time-varying $\mathbf{x}_o := \mathbf{x}_n$, and parameters $P = 30$, $P_1 = 3$, $v_n \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = 10^{-1}$, and $\mathbf{h}_n \sim \mathcal{N}\left(\mathbf{0}_P, \frac{1}{P}\mathbf{I}_P\right)$. A Gauss-Markov model is assumed for $\mathbf{x}_n$, that is $x_n(p) = \alpha x_{n-1}(p) + w_n(p)$ with $x_0(p) \sim \mathcal{N}(0,1)$, $\alpha = 0.99$, and $w_n(p) \sim \mathcal{N}\left(0, 1-\alpha^2\right)$ for $p = 1, 2, 3$. Without loss of generality, (c2) is adopted with $\beta = 0.9$, and penalty parameter $\lambda_N = \sqrt{2\sigma^2 \log P}\sqrt{\sum_{n=1}^{N} \beta^{2(N-n)}}$. Figure 2 depicts the true variation of $x_n(1)$ across time along with those for the TW-Lasso and the OSCD TW-Lasso estimates. Observe that OSCD closely tracks TW-Lasso iterates and both estimators remain close to the true signal.

To test the ability of OSCD TW-Lasso in estimating accurately zero entries of $\mathbf{x}_o$, Fig. 3 shows the RLS, TW-Lasso and OSCD TW-Lasso estimates of $x_o(4) = 0$ across time. While RLS estimates fluctuate widely, the TW-Lasso and OSCD TW-Lasso estimate accurately this inactive entry at zero.

5. CONCLUSIONS

Low-complexity sparsity-aware recursive schemes were developed for adaptive filtering. At the heart of these schemes is a novel optimization algorithm that implements the coordinate descent approach online. Coordinate-wise optimization of Lasso-like problems can be expressed in closed form, which facilitates implementation and numerical stability. Albeit simple, the OCD TW-Lasso is provably convergent when the sparse signal is time invariant. At complexity comparable to OCD but with improved convergence speed, the OSCD TW-Lasso selects the *best* component to optimize and exhibits performance similar to TW-Lasso, which can be prohibitively complex to implement for real time applications.

6. REFERENCES

[1] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. R. Stat. Soc. Ser. B*, vol. 58, no. 1, pp. 267–288, 1996.

[2] E. J. Candes and M. Wakin, "An introduction to compressive sampling," *IEEE Sign. Proc. Mag.*, vol. 25, pp. 21–30, March 2008.

[3] S. Sanghavi, D. M. Malioutov and A. S. Willsky, "Compressed sensing with sequential observations," in *Proc. IEEE Intl. Conf. Ac. Sp. Sig. Proc.*, Las Vegas, NV, Apr. 2008.

[4] Y. Gu Y. Chen and A. O. Hero, "Sparse LMS for system identification," in *Proc. IEEE Intl. Conf. Ac. Sp. Sig. Proc.*, Taipei, Taiwan, Apr. 2009.

[5] A. H. Sayed, *Fundamentals of Adaptive Filtering*, John Wiley & Sons, 2003.

[6] D. Angelosante and G. B. Giannakis, "RLS-weighted Lasso for adaptive estimation of sparse signals," in *Proc. IEEE Intl. Conf. Ac. Sp. Sig. Proc.*, Taipei, Taiwan, Apr. 2009.

[7] J. Friedman, T. Hastie, H. Höfling and R. Tibshirani, "Pathwise coordinate optimization," *The Annals of Applied Statistics*, vol. 1, pp. 302–332, Dec. 2007.

[8] T. T. Wu and K. Lange, "Coordinate descent algorithms for Lasso penalized regression," *The Annals of Applied Statistics*, vol. 2, pp. 224–244, March 2008.

[9] P. Tseng, "Convergence of block coordinate descent method for nondifferentiable minimization," *J. Optim. Theory Appl.*, vol. 109, pp. 475–494, June 2001.

[10] G. P. White, Y. V. Zakharov and J. Liu, "Low-complexity RLS algorithms using dichotomous coordinate descent iterations," *IEEE Trans. Sign. Proc.*, vol. 56, pp. 3150–3161, July 2008.