SPARSE LMS VIA ONLINE LINEARIZED BREGMAN ITERATION

Tao Hu[1] *and Dmitri B. Chklovskii*[2]

[1]Center for Bioinformatical and Genomic Systems Eng, Texas A&M, College Station, TX, USA
Email: taohu@tees.tamus.edu
[2]Howard Hughes Medical Institute, Janelia Farm Research Campus, Ashburn, VA, USA
Email:chklovskiid@janelia.hhmi.org

ABSTRACT

We propose a version of least-mean-square (LMS) algorithm for sparse system identification. Our algorithm called online linearized Bregman iteration (OLBI) is derived from minimizing the cumulative prediction error squared along with an l_1-l_2 norm regularizer. By systematically treating the non-differentiable regularizer we arrive at a simple two-step iteration. We demonstrate that OLBI is bias free and compare its operation with existing sparse LMS algorithms by rederiving them in the online convex optimization framework. We perform convergence analysis of OLBI for white input signals and derive theoretical expressions for the steady state mean square deviations (MSD). We demonstrate numerically that OLBI improves the performance of LMS type algorithms for signals generated from sparse tap weights.

Index Terms— *LMS, Sparse, online*

1. INTRODUCTION

Many signal processing tasks, such as signal prediction, noise cancellation or system identification, reduce to predicting a time varying signal, f_k [1]. In many cases such prediction can be computed as a linear combination of the input signal vector, $\boldsymbol{x}_k$, with the weight vector, $\boldsymbol{w}_k$. The goal is to minimize the cumulative empirical loss defined as the square of the prediction error,

$$\sum_{k=1}^{t} \ell_k(f_k, \boldsymbol{w}_k) = \sum_{k=1}^{t} \frac{1}{2}(f_k - \boldsymbol{w}_k^T \boldsymbol{x}_k)^2, \quad (1)$$

by continuously adjusting the weights, $\boldsymbol{w}_k$, based on the previous prediction error. This adaptive filtering framework, Fig. 1, can be formalized as follows:

Initialize the weight vector, $\boldsymbol{w}_1$. At each time step $k = 1,2,\dots,t$

- Get input signal vector, $\boldsymbol{x}_k = [x_{k,0}, x_{k,1}, \dots, x_{k,n-1}]^T$.
- Compute a filter output, $\boldsymbol{w}_k^T \boldsymbol{x}_k$, for the desired output, f_k, using the weight vector, $\boldsymbol{w}_k = [w_{k,0}, w_{k,1}, \dots, w_{k,n-1}]^T$.
- Observe desired output signal, f_k.
- Adjust weight vector $\boldsymbol{w}_k \Rightarrow \boldsymbol{w}_{k+1}$, using prediction error, $f_k - \boldsymbol{w}_k^T \boldsymbol{x}_k$.

A popular algorithms for adjusting filter weights is the least-mean-square (LMS) [1]:

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \delta(f_k - \boldsymbol{w}_k^T \boldsymbol{x}_k)\boldsymbol{x}_k, \quad (2)$$

where δ is the update step size. The popularity of LMS comes from the simplicity of implementation and robustness of performance [1].

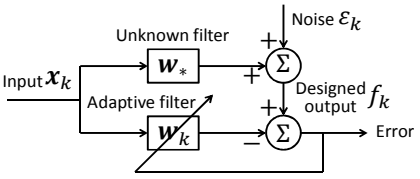

Fig. 1. Adaptive filter.

The performance of the LMS algorithm can be improved if additional information is available about the statistical properties of the true weight vector, $\boldsymbol{w}_*$ ($f_k = \boldsymbol{w}_*^T \boldsymbol{x}_k + \varepsilon_k$, where the observation noise ε_k, is assumed to be independent of $\boldsymbol{x}_k$). In particular, if the true weight vector is sparse, i.e. contains only a few non-zero weights, the LMS algorithm can be modified to reflect such prior knowledge [2-15]. Specifically, a class of LMS algorithms [11-15] incorporates the sparse prior by introducing a sparsity-inducing l_p-norm of the weight vector to the loss function (1) and demonstrates better performance than (2).

Here, we show that the l_p-norm constrained algorithms [11-15] can be derived in the online convex optimization (OCO) framework [16-18], which was previously used to derive the standard LMS algorithm (2) in order to account for its robustness [19]. Inspired by recently developed sparse online learning methods [20, 21], we use the OCO framework to derive a novel algorithm for predicting signals generated from sparse weights, which we call online linearized Bregman iteration (OLBI). Compared to the existing algorithms [11-15], the derivation of OLBI relies on a systematic treatment of non-differentiable cost function [22-24] using sub-differentials as is done in convex optimization theory [25]. Although invoking sub-differentials leads to the appearance of an additional step in OLBI compared to standard LMS, it remains computationally efficient. We prove analytically that the steady state performance of OLBI exceeds that of LMS for

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

sparse weight vectors and compare the performance of OLBI numerically to that of recent sparse LMS modifications.

The paper is organized as follows. We start with reviewing the derivation the standard LMS from the OCO framework in Section 2. In Section 3, we derive OLBI from the OCO framework by using the l_1-l_2 norm as the sparsity-inducing regularizer. In Section 4 we compare OLBI and existing sparse LMS algorithms in the OCO framework. In Section 5 we summarize the results of convergence analysis. Theorems are given without proof due to space constraints. We demonstrate the performance of OLBI relative to existing algorithms using numerical simulations in Section 6. Finally, in Section 7, we conclude and discuss possible future improvements of OLBI.

2. DERIVATION of STANDARD LMS from OCO

We start by deriving standard LMS following the OCO framework [16-18]. We update $\boldsymbol{w}_{k+1}$ by minimizing the cost function comprised of a cumulative loss, $\sum_{s=1}^{k} \ell_s(f_s, \boldsymbol{w})$, and a regularizer, $\Psi(\boldsymbol{w})$:

$$\boldsymbol{w}_{k+1} = \operatorname{argmin}_{\boldsymbol{w}}\{\delta \sum_{s=1}^{k} \ell_s(f_s, \boldsymbol{w}) + \Psi(\boldsymbol{w})\}, \quad (3)$$

where $\delta > 0$ will turn out to be the learning rate. The first prediction is generated by initializing the weights, $\boldsymbol{w}_1 = \operatorname{argmin}_{\boldsymbol{w}}\{\Psi(\boldsymbol{w})\}$. Therefore, $\Psi(\boldsymbol{w})$ can be thought to represent our knowledge about $\boldsymbol{w}$ prior to the first time step. The convex cost function ℓ_s is usually approximated by a linear form of its gradients [18], $g_k(\boldsymbol{w}_k) = \nabla l_k(\boldsymbol{w}_k)$ and (3) becomes

$$\boldsymbol{w}_{k+1} = \operatorname{argmin}_{\boldsymbol{w}}\{\delta \sum_{s=1}^{k} \langle g_s(\boldsymbol{w}_s), \boldsymbol{w} - \boldsymbol{w}_s\rangle + \Psi(\boldsymbol{w})\}. \quad (4)$$

To derive standard LMS, we choose the regularizer as the l_2-norm of the filter weights,

$$\Psi(\boldsymbol{w}) = \tfrac{1}{2}\|\boldsymbol{w}\|_2^2. \quad (5)$$

Furthermore, we choose the loss at each time step, ℓ_s, as the prediction error squared and, therefore,

$$g_s(\boldsymbol{w}_s) = \nabla \ell_s(\boldsymbol{w}_s) = -(f_s - \boldsymbol{w}_s^T \boldsymbol{x}_s)\boldsymbol{x}_s. \quad (6)$$

Substituting (5) and (6) into (4), we obtain

$$\boldsymbol{w}_{k+1} = \delta \sum_{s=1}^{k} (f_s - \boldsymbol{w}_s^T \boldsymbol{x}_s)\boldsymbol{x}_s. \quad (7)$$

Similarly, we obtain $\boldsymbol{w}_k = \delta \sum_{s=1}^{k-1} (f_s - \boldsymbol{w}_s^T \boldsymbol{x}_s)\boldsymbol{x}_s$. After substituting this expression in place of the first k-1 terms of the sum in (7), we find the LMS update rule (2). If we assume that the true filter $\boldsymbol{w}_*$ is constant, and that the input signal $\boldsymbol{x}_k$ is wide-sense stationary, then $\boldsymbol{w}_k$ converges to $\boldsymbol{w}_*$, $\lim_{k\to\infty} E[\boldsymbol{w}_k] = \boldsymbol{w}_*$, if and only if $0 < \delta < 1/\lambda_{\max}$ [1]. Here $\lambda_{\max}$ is the greatest eigenvalue of the input covariance matrix $\boldsymbol{C} = E[\boldsymbol{x}_k \boldsymbol{x}_k^T]$. Therefore the standard LMS algorithm is bias free.

Below, we use a similar approach, along with the sparsity prior on the weights, to derive OLBI.

3. DERIVATION of OLBI

To derive OLBI, we use a regularizer, $\Psi(\boldsymbol{w}) = \gamma\|\boldsymbol{w}\|_1 + \frac{1}{2}\|\boldsymbol{w}\|_2^2$, which is the l_1-l_2 norm of the filter weights also known as the elastic net [26]. By substituting this regularizer in Eq. (4) we obtain:

$$\boldsymbol{w}_{k+1} = \operatorname{argmin}_{\boldsymbol{w}}\{\delta \sum_{s=1}^{k} \langle g_s(\boldsymbol{w}_s), \boldsymbol{w} - \boldsymbol{w}_s\rangle + \gamma\|\boldsymbol{w}\|_1 + \tfrac{1}{2}\|\boldsymbol{w}\|_2^2\}. \quad (8)$$

The optimality condition for the weights in (8) is:

$$\boldsymbol{m}_{k+1} \in \partial\big[\gamma\|\boldsymbol{w}_{k+1}\|_1 + \tfrac{1}{2}\|\boldsymbol{w}_{k+1}\|_2^2\big], \quad (9)$$

where $\partial[.]$ designates a sub-differential [25] and $-\boldsymbol{m}_{k+1}$ is the gradient of the first term in (8):

$$\boldsymbol{m}_{k+1} = -\delta \sum_{s=1}^{k} g_s(\boldsymbol{w}_s). \quad (10)$$

Similarly, from the condition of optimality for w_k

$$\boldsymbol{m}_k = -\delta \sum_{s=1}^{k-1} g_s(\boldsymbol{w}_s). \quad (11)$$

By combining (10) and (11), we get

$$\boldsymbol{m}_{k+1} = \boldsymbol{m}_k - \delta g_k(\boldsymbol{w}_k). \quad (12)$$

Substituting (10) into (8) and simplifying, we obtain

$$\boldsymbol{w}_{k+1} = \operatorname{argmin}_{\boldsymbol{w}}\big\{-\langle \boldsymbol{m}_{k+1}, \boldsymbol{w} - \boldsymbol{w}_k\rangle + \gamma\|\boldsymbol{w}\|_1 + \tfrac{1}{2}\|\boldsymbol{w}\|_2^2\big\} = \operatorname{argmin}_{\boldsymbol{w}}\big\{\tfrac{1}{2}\|\boldsymbol{w} - \boldsymbol{m}_{k+1}\|_2^2 + \gamma\|\boldsymbol{w}\|_1\big\}. \quad (13)$$

Such minimization problem can be solved component-wise using a shrinkage (soft-thresholding) operation [27, 28],

$$\operatorname{shrink}(a, \gamma) = \begin{cases} a - \gamma, & \text{if } a > \gamma \\ 0, & \text{if } -\gamma \le a \le \gamma \\ a + \gamma, & \text{if } a < -\gamma \end{cases}. \quad (14)$$

By combining (6), (12) and (13), we arrive at the following:

Algorithm 1 Online linearized Bregman iteration (OLBI)
initialize: $\boldsymbol{m}_1$=0 and $\boldsymbol{w}_1$=0.
for k=1,2,3,…**do**

$$\boldsymbol{m}_{k+1} = \boldsymbol{m}_k + \delta(f_k - \boldsymbol{w}_k^T \boldsymbol{x}_k)\boldsymbol{x}_k, \quad (15)$$

$$\boldsymbol{w}_{k+1} = \operatorname{shrink}(\boldsymbol{m}_{k+1}, \gamma), \quad (16)$$

end for

The reason for the algorithm name becomes clear if we substitute (10) and (12) into (8) and obtain:

$$\boldsymbol{w}_{k+1} = \operatorname{argmin}_{\boldsymbol{w}}\{\delta\langle g_k(\boldsymbol{w}_k), \boldsymbol{w} - \boldsymbol{w}_k\rangle + \gamma\|\boldsymbol{w}\|_1 + \tfrac{1}{2}\|\boldsymbol{w}\|_2^2 - \langle \boldsymbol{m}_k, \boldsymbol{w} - \boldsymbol{w}_k\rangle\}, \quad (17)$$

which can be written as:

$$\boldsymbol{w}_{k+1} = \operatorname{argmin}_{\boldsymbol{w}}\big\{\delta\langle g_k(\boldsymbol{w}_k), \boldsymbol{w} - \boldsymbol{w}_k\rangle + D_{\Psi}^{\boldsymbol{m}_k}(\boldsymbol{w}, \boldsymbol{w}_k)\big\}, \quad (18)$$

where

$$D_{\Psi}^{\boldsymbol{m}_k}(\boldsymbol{w}, \boldsymbol{w}_k) = \Psi(\boldsymbol{w}) - \Psi(\boldsymbol{w}_k) - \langle \boldsymbol{m}_k, \boldsymbol{w} - \boldsymbol{w}_k\rangle \quad (19)$$

is the Bregman divergence induced by the convex function $\Psi(\boldsymbol{w})$ at point $\boldsymbol{w}_k$ for sub-gradient $\boldsymbol{m}_k$ [29]. In fact, (18) can be a starting point for the derivation of OLBI, thus justifying the name. Previously in [22-24], a related algorithm called linearized Bregman iteration was proposed for solving deterministic convex optimization problems, such as basis pursuit [30].

4. DERIVATION of OTHER SPARSE LMS ALGORITHMS

In this section, we briefly review recent sparse LMS algorithms [11-14] which are closely related to our algorithm. We demonstrate that all these algorithms can be derived from the OCO framework with slightly different

choices of the loss function and the regularization function in (3).

4.1 ZA-LMS and RZA-LMS

To derive ZA-LMS [11, 12] we start from Eq. (4) but modifying the loss function by adding an l_1 term:

$$\ell_s^{\mathrm{ZA}}(f_s, \boldsymbol{w}_s) = \frac{1}{2}(f_s - \boldsymbol{w}_s^T \boldsymbol{x}_s)^2 + \rho\|\boldsymbol{w}_s\|_1, \quad (20)$$

while keeping the same l_2-norm regularizer $\Psi(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2$. Then we write the weight update as:

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \delta\nabla\ell_s^{\mathrm{ZA}}(f_s, \boldsymbol{w}_s). \quad (21)$$

Since the l_1-norm is not differentiable at zero, following [11, 12], we set the gradient at zero to be zero, which is the value of one of its subgradients. Then we obtain the update rule of ZA-LMS

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \delta(f_k - \boldsymbol{w}_k^T \boldsymbol{x}_k)\boldsymbol{x}_k + \delta\rho h^{\mathrm{ZA}}(\boldsymbol{w}_k), \quad (22)$$

where the component-wise zero-point attraction function is given by

$$h^{\mathrm{ZA}}(z) = -\mathrm{sgn}(z) = \begin{cases} \frac{z}{|z|} & z \neq 0 \\ 0 & z = 0 \end{cases}. \quad (23)$$

To restrict attraction to smaller weight values, in the reweighted ZA-LMS or RZA-LMS algorithm [11, 12], the uniform attractor (23) is replaced by:

$$h^{\mathrm{RZA}}(z) = -\mathrm{sgn}(z)/(1 + \varepsilon|z|), \quad (24)$$

which generates weaker attraction for larger weights. Replacing $h^{\mathrm{ZA}}(z)$ in (22) by $h^{\mathrm{RZA}}(z)$ yields the update rule for RZA-LMS, which can be derived from the OCO framework, (4) by using the following loss function:

$$\ell_s^{\mathrm{RZA}}(f_s, \boldsymbol{w}_s) = \frac{1}{2}(f_s - \boldsymbol{w}_s^T \boldsymbol{x}_s)^2 + \rho\sum_i \log\left(1 + \frac{|w_{s,i}|}{\varepsilon}\right). \quad (25)$$

4.2 l_0-LMS

We can derive l_0-LMS [13, 14] by using a loss function with an additional l_0-norm

$$\ell_s^{l_0}(f_s, \boldsymbol{w}_s) = \frac{1}{2}(f_s - \boldsymbol{w}_s^T \boldsymbol{x}_s)^2 + \kappa\|\boldsymbol{w}_s\|_0, \quad (26)$$

Similar to the ZA-LMS and RZA-LMS loss functions (20, 25), the l_0-LMS loss function (26) is not differentiable at zero. Following [13, 14], we approximate the non-differentiable l_0-norm as a sum of exponentials $\|\boldsymbol{w}_s\|_0 \approx \sum_i\left[1 - \exp\left(-\alpha|w_{s,i}|\right)\right]$ [31], take the first order term of the Taylor expansion and set the gradient at zero to be zero, thus obtaining

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \delta(f_k - \boldsymbol{w}_k^T \boldsymbol{x}_k)\boldsymbol{x}_k + \delta\kappa\sum_i\left[\alpha^2 w_{s,i} - \alpha\mathrm{sgn}\left(w_{s,i}\right)\right]. \quad (27)$$

To achieve zero-point attraction, the sign of $w_{s,i}\left[\alpha^2 w_{s,i} - \alpha\mathrm{sgn}\left(w_{s,i}\right)\right]$ should be non-positive. Therefore, as in [13, 14], we replace the last term in (27) by the following zero-point attraction function, $h^{l_0}(\boldsymbol{w}_k)$, which shrinks the small coefficients in the interval $[-1/\alpha, 1/\alpha]$ towards zero:

$$h^{l_0}(z) = \begin{cases} \alpha^2 z - \alpha\mathrm{sgn}(z) & |z| \leq 1/\alpha \\ 0 & \text{elsewhere} \end{cases}, \quad (28)$$

and arrive at the l_0-LMS update rule [13, 14]:

$$\boldsymbol{w}_{k+1} = \boldsymbol{w}_k + \delta(f_k - \boldsymbol{w}_k^T \boldsymbol{x}_k)\boldsymbol{x}_k + \delta\kappa h^{l_0}(\boldsymbol{w}_k). \quad (29)$$

Because we were able to derive OLBI and other sparse LMS algorithms in the common OCO framework, we are now in a position to compare their operation. First, unlike OLBI where sparsity inducing terms are added to the regularizer, in ZA-LMS, RZA-LMS and l_0-LMS they are added to the loss function. Because of the zero-point attraction function, the steady state solution in algorithms [11-14] is biased, i.e. $\lim_{k\to\infty} E[\boldsymbol{w}_k] \neq \boldsymbol{w}_*$. In contrast, OLBI is derived by adding the sparsity-inducing terms to the regularizer and introducing a second set of weights related by a continuous shrinkage operation (14). This yielded a two-step iteration, which does not attract the filter weights to zeros directly and generates sparse solution without causing bias in the steady state (Theorem 1).

Second, instead of approximating the non-differentiable sparsity inducing functions or heuristically setting the gradient to zero where it does not exist, OLBI handles the non-differentiability properly using sub-gradients [25].

5. CONVERGENCE ANALYSIS

In this section, we summarize the main analytical results of convergence. Due to space constraints, theorems are given without proof. We assume that the true filter weight vector $\boldsymbol{w}_*$ is constant, and that the input $\boldsymbol{x}_k$ is an *i.i.d.* zero-mean Gaussian signal. We also assume that the zero-mean additive noise ε_k is independent of $\boldsymbol{x}_k$.

Theorem 1. (Mean performance) If $0 < \delta < \frac{1}{\lambda_{\max}}$, then $\lim_{k\to\infty} E[\boldsymbol{w}_k] = \boldsymbol{w}_*$, where $\lambda_{\max}$ is the maximum eigenvalue of the input covariance matrix $\boldsymbol{C} = E[\boldsymbol{x}_k \boldsymbol{x}_k^T]$.

Theorem 2. (Steady state mean square deviation (MSD)) If $0 < \delta < \frac{2}{(\|\boldsymbol{w}_*\|_0 + 2)\sigma_x^2}$, then algorithm 1 converges in the mean square sense and the steady state mean square deviation is

$$D_\infty^{OLBI} = \lim_{k\to\infty} E[\boldsymbol{\xi}_k^T \boldsymbol{\xi}_k] = \frac{\delta\sigma_\varepsilon^2\|\boldsymbol{w}_*\|_0}{2 - \delta\sigma_x^2(\|\boldsymbol{w}_*\|_0 + 2)}, \quad (30)$$

where the l_0-norm $\|\boldsymbol{w}_*\|_0$ counts the number of non-zero entries in $\boldsymbol{w}_*$.

Compared to the steady state MSD of standard LMS [1]

$$D_\infty^{LMS} = \frac{\delta\sigma_\varepsilon^2 n}{2 - \delta\sigma_x^2(n + 2)}, \quad (31)$$

in the OLBI result (30), the filter length n is replaced by the number of non-zeros entries. At the limit of small learning rate,

$$D_\infty^{OLBI}/D_\infty^{LMS} = \|\boldsymbol{w}_*\|_0/n. \quad (32)$$

Therefore, for a sparse filter with $\|\boldsymbol{w}_*\|_0/n \ll 1$, OLBI achieves a much smaller steady state MSD.

6. NUMERICAL RESULTS

In this section, we perform numerical simulations to confirm the theoretical results and to demonstrate the effectiveness of

OLBI. We consider white input signal $\{x_k\}$ drawn from Gaussian distribution with zero mean and unit variance. The measurement noise $\{\varepsilon_k\}$ is also a zero-mean white Gaussian process with variance σ_ε^2 adjusted to achieve SNR=20dB or 10dB. The filter length is set to be $n = 1000$, and the filter sparsity $\|\boldsymbol{w}_*\|_0$ varies from 50 to 1000. The non-zero coefficients in $\boldsymbol{w}_*$ are drawn from a Gaussian distribution with zero mean and unit variance and their locations are randomly assigned. In all numerical experiments, unless mentioned otherwise, the learning rate $\delta = 8 \times 10^{-4}$, the threshold $\gamma = 0.5$, and the filter sparsity is set to 100. The results are shown as averages of 100 independent trials.

First, we consider the steady state MSD of OLBI with respect to the learning rate δ. Figure 2 shows that the simulation results are consistent with our analytical results.

Next, we investigate the dependence of steady state MSD on the threshold γ. The simulation results follow the theoretical predictions well, Fig. 3. When the threshold is large enough, both the steady state MSD (Fig. 3a and c) and the solution sparsity $\|\boldsymbol{w}\|_0$ (Fig. 3b and d) do not depend on γ. But when γ is small, because of noise, zero entries in $\boldsymbol{w}_*$ frequently cross the threshold, yielding solution sparsity and MSD different from true filter sparsity and the theoretical result. Yet, compared to standard LMS, OLBI still achieves smaller MSD. When $\gamma = 0$, OLBI is identical to standard LMS, which is exemplified by the same MSD and solution sparsity $\|\boldsymbol{w}\|_0$, Fig. 3.

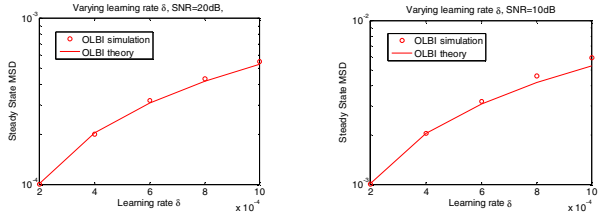

Fig. 2. Dependence of steady state MSD on learning rate δ for SNR=20dB, (a) and 10dB, (b).

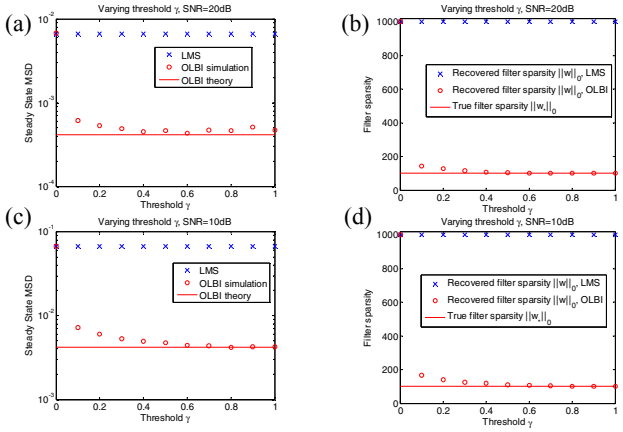

Fig. 3. Dependence of steady state MSD and filter sparsity $\|w\|_0$ on threshold λ for SNR=20dB, (a) and (b); SNR=10dB, (c) and (d).

In the third experiment, we evaluate the steady state MSD with respect to the system sparsity, Fig. 4. We also included in the comparison other sparse LMS algorithms l_0-LMS, ZA-LMS and RZA-LMS with optimized parameters [11-14]. We found the OLBI shows better steady state performance in the tested parameter range thus confirming our theoretical analysis.

The last experiment is designed to investigate the convergence properties of OLBI for different thresholds, γ, Fig. 5. Smaller thresholds reduce time needed to cross threshold and thus faster convergence rate, but at the expense of denser solutions and larger steady state MSD. When the threshold is large enough, the steady state MSD loses its dependence on γ. Increasing γ only reduces the convergence rate without decreasing the steady state MSD.

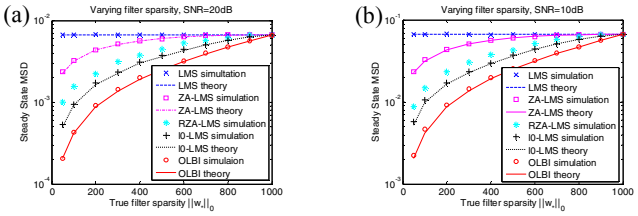

Fig. 4. Dependence of steady state MSD on true filter sparsity for SNR=20dB, (a) and 10dB, (b).

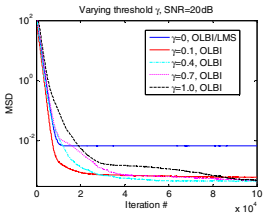

Fig. 5. Instantaneous MSD of standard LMS and OLBI with respect to different thresholds.

7. CONCLUSION

We presented a new algorithm called OLBI for solving sparse adaptive signal processing problems. OLBI can be viewed as a sparse version of LMS, which achieves both sparse and unbiased steady state solution. We compared the operation of OLBI with existing sparse LMS algorithms by re-deriving them in the OCO framework. We analyzed its mean square performance. With reasonable approximations, we derived the theoretically results for the steady state MSD. We showed both theoretically and numerically that OLBI is superior to standard LMS for sparse system identification. OLBI also enjoys both low computational complexity and low storage requirement. We may further improve the performance of OLBI by adaptively adjust the learning rate δ, the shrinkage threshold γ and incorporate the prior information of structured sparsity for some filters.

References

[1] Widrow, B. & Stearns, S.D. (1985) *Adaptive Signal Processing*. New Jersey: Prentice Hall.

[2] Benesty, J. & Gay S. L. (2002) An Improved PNLMS algorithm. *Proc. ICASSP*, pp. 1881-1884.

[3] Godavarti, M. & Hero, A.O. (2005) Partial update LMS algorithms. *IEEE Trans. Signal Process.* **53**(7):2382-2399.

[4] Abadi, M. & Husoy, J. (2008) Mean-square performance of the family of adaptive filters with selective partial updates. *Signal Processing*, **88**(8):2008-2018.

[5] Arenas-Garcia, J. & Figueiras-Vidal, A.R. (2008) Adaptive combination of IPNLMS filters for robust sparse echo cancellation. *Proc. MLSP*, pp. 221-226.

[6] Deng, H. & Dyba, R.A. (2009) Partial update PNLMS algorithm for network echo cancellation. *ICASSP*, pp. 1329-1332.

[7] Murakami, Y., Yamagishi, M., Yukawa, M. & Yamada, I. (2010) A sparse adaptive filtering using time-varying soft-thresholding techniques. *ICASSP*, pp. 3734-3737.

[8] Yang, J. & Sobelman, G.E. (2010) Sparse LMS with segment zero attractors for adaptive estimation of sparse signals. *APCCAS*, pp. 422-425.

[9] Shi, K. & Ma, X. (2010) Transform domain LMS algorithms for sparse system identification. *Proc. ICASSP*, pp. 3714-3717.

[10] Kopsinis, Y., Slavakis, K., Theodoridis, S. & McLaughlin, S. (2011) Online sparse system identification and signal reconstruction using projections onto weighted l_1-balls, *IEEE Trans. Signal Proc.,* **59**:936-952.

[11] Chen, Y., Gu, Y. & Hero, A.O. Regularized least-mean-square algorithms. Available: http://arxiv.org/pdf/1012.5066.

[12] Chen, Y., Gu, Y. & Hero, A.O. (2009) Sparse LMS for system identification, *Proc. ICASSP*, pp. 3125-3128.

[13] Gu, Y., Jin, J. & Mei, S. (2009) l0 Norm constraint LMS algorithm for sparse system identification. *IEEE Signal Process. Lett.* **16**(9):774-777.

[14] Su, G., Jin, J., Gu, Y. & Wang, J. Performance Analysis of l0 Norm Constraint Least Mean Square Algorithm. Available: http://arxiv.org/pdf/1203.1535.pdf.

[15] Jin, J., Qu, Q. & Gu, Y. Robust zero-point attraction least mean square algorithm on near sparse system identification. IET Signal Processing **7**(3):210-218.

[16] Cesa-Bianchi, N., Long, P.M. & Warmuth, M.K. (1996) Worst-case quadratic loss bounds for a generalization of the Widrow-Hoff rule. *IEEE Trans. Neural Networks*, **7**:604–619.

[17] Cesa-Bianchi, N. & Lugosi, G. (2006) *Prediction, Learning, and Games*. Cambridge: Cambridge University Press.

[18] Rakhlin, A. (2009) Lecture Notes on Online Learning. Available: http://www-stat.wharton.upenn.edu/~rakhlin/courses/stat991/papers/lecture_notes.pdf.

[19] Hassibi, B. (2003) On the robustness of LMS filters, in Least-Mean-Square Adaptive Filters, Haykin, S. & Widrow, B. Eds., John Wiley & Sons.

[20] Xiao, L. (2010) Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, **9**:2543-2596.

[21] Langford, J., Li, L. & Zhang, T. (2009) Sparse online learning via truncated gradient. *Journal of Machine Learning Research*, **10**:777-801.

[22] Cai, J.F., Osher, S. & Shen, Z. (2009) Convergence of the linearized Bregman iteration for l_1-norm minimization. *Mathematics of Computation,* **78**:2127-2136.

[23] Cai, J.F., Osher, S. & Shen, Z. (2009) Linearized Bregman iterations for compressed sensing. *Mathematics of Computation,* **78**:1515-1536.

[24] Yin, W., Osher, S., Goldfarb, D. & Darbon, J. (2008) Bregman iterative algorithms for l_1-minimization with applications to compressed sensing. *Siam Journal on Imaging Sciences,* **1**:143-168.

[25] Clarke, F. H. (1990) *Optimization and Nonsmooth Analysis*. SIAM, Philadelphia.

[26] Zou, H. & Hastie, T. (2005) Regularization and variable selection via the elastic net. *Journal of the Royal Statistical* Society *Series B-Statistical Methodology,* **67**:301-320.

[27] Daubechies, I., Defrise, M. & De Mol, C. (2004) An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied* Mathematics, **57**:1413-1457.

[28] Elad, M., Matalon, B., Shtok, J. & Zibulevsk, M. (2007) A wide-angle view at iterated shrinkage algorithms. *Proc. SPIE*, pp. 26-29.

[29] Bregman, L.M. (1967) The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, **7**:200-217.

[30] Chen, S.S., Donoho, D.L. & Saunders, M.A. (1998) Atomic decomposition by basis pursuit. *Siam Journal on Scientific Computing,* **20**:33-61.

[31] Weston, J., Elisseeff, A., Scholkopf, B. & Tipping, M. (2003) Use of the zero norm with linear models and kernel methods. *Journal of Machine Learning Research*, **3**:1439-1461.