https://doi.org/10.1016/j.sigpro.2016.04.013

A New QR Decomposition-based RLS Algorithm Using the Split Bregman Method for L_1-Regularized Problems

Y. J. Chu and C. M. Mak

***Abstract*—The split Bregman (SB) method can solve a broad class of L_1-regularized optimization problems and has been widely used for sparse signal processing in a variety of applications. To achieve lower complexity and to cope with time-varying environments, we develop a new adaptive version of the SB method for finding online sparse solutions. This algorithm is derived from the recursive least squares (RLS) optimization problem, where the SB method is used to separate the regularization term from the constrained optimization. This algorithm is numerically more stable and easily amenable to multivariate implementation due to the use of a QR decomposition (QRD) structure. An efficient method is further developed for selecting the thresholding rule, which controls the sparsity level of the estimator. Moreover, the SB-QRRLS algorithm is extended to a multivariate version to solve the sparse principal component analysis (SPCA) problem. Simulation results are presented to illustrate the effectiveness of the proposed algorithms in sparse system estimation and SPCA. We show that the convergence and tracking performance of the proposed algorithms compares favorably with conventional algorithms.**

***Keywords*—L_1 regularization, split Bregman (SB), recursive least squares (RLS), sparse principal component analysis (SPCA).**

Y. J. Chu and C. M. Mak are with the Department of Building Services Engineering, Hong Kong Polytechnic University, Hong Kong (e-mail: yijing.chu@polyu.edu.hk; cheuk-ming.mak@polyu.edu.hk).

I. INTRODUCTION

L_1-regularized problems have received much attention recently due to the use of compressed sensing (CS) in a variety of applications, such as image restoration and biomedical signal processing. To solve these optimization problems, the split Bregman (SB) method [1] is frequently employed, where the constrained optimization problems can be reduced to a short sequence of unconstrained least squares (LS) problems using the Bregman iteration [2]. Compared to conventional Newton-type methods, the SB method has much faster convergence and higher numerical efficiency. Most of the SB-based algorithms are in batch form and iterative methods are frequently employed. These algorithms, however, are usually computationally consuming and are not quite amenable to online applications in time-varying environments [3].

To overcome these difficulties, we propose a new adaptive SB algorithm for finding sparse solutions to the LS problem. Adaptive estimation for sparse signals has received little attention until recently. One challenge arises from the gradient-based minimization of the LS cost function, which is usually impossible due to the non-differentiable L_1 norm. A possible solution is offered by the weighted L_2 norm in [4]. Convex relaxation techniques, such as coordinate descent and expectation maximization, have also been used for adapting the filter coefficients [5]-[9]. Another algorithm, namely the greedy RLS (GRLS) algorithm [10], uses orthogonal LS to compute a true sparse LS solution and employs information-theoretic criteria for selecting the number of nonzero elements. For time-varying channels, the GRLS algorithm usually has better performance than others due to the information-theoretic criteria used, but simplified versions with lower-complexity are desired.

In this paper, we propose a new adaptive approach for L_1-regularized problems by applying the SB method to the QR decomposition (QRD)-based RLS algorithm, and call this the SB-QRRLS algorithm. To our best knowledge, this is the first time that SB is used in an adaptive filter to compute sparse solutions. First, an exponential window is applied to the L_1-regularized optimization problem, and a recursive SB algorithm is derived. The resulting cost function for the algorithm, including weighted square error and regularization of the split variable, is then minimized adaptively using QRD [11]. This QRD structure is used for its numerical efficiency [12] in solving regularized LS problems and lower roundoff error. In such RLS-like algorithms, the regularization parameter controls the sparsity level and thus plays an important role in balancing between the estimation variance and undesirable bias introduced by L_1 regularization. In this short paper, the effect of the regularization parameter on the performance of the SB-QRRLS algorithm is studied and a formula for selecting the regularization parameter is provided. The proposed SB-QRRLS algorithm is then extended to its multivariate version for the sparse principal component analysis (SPCA) problem [13]; we have found no similar method in the literature.

The rest of the letter is organized as follows: in Section II, the SB-QRRLS algorithm is derived and the selection of the regularization parameter is discussed. Section III is devoted to SPCA using a multivariate version of the SB-QRRLS. The validity of the proposed algorithms is examined in Section IV. Finally, conclusions are drawn in Section V.

II. The Proposed SB-QRRLS Algorithm

A. SB Weight Update

Many important problems in imaging science, gene networks, and other computational areas can be posed as L_1-regularized LS optimization problems with solutions of the form

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} [\tfrac{\xi}{2} \| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} \|_2^2 + \Phi(\boldsymbol{\Theta}\boldsymbol{w})] \tag{1}$$

where $\boldsymbol{y} = [y_1, y_2 ... y_N]^T$ is the observation vector of dimension N, $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2 ... \boldsymbol{x}_N]^T$ is the $N \times L$ system matrix, Φ is a convex potential function, e.g. the frequently-used L_1 norm $\Phi(\boldsymbol{u}) = \| \boldsymbol{u} \|_1$, $\boldsymbol{\Theta}$ is an analysis matrix, and ξ is a positive constant serving as the L_1 regularization parameter. Applying Bregman splitting to (1) yields

$$(\hat{\boldsymbol{w}}, \hat{\boldsymbol{v}}) = \arg\min_{\boldsymbol{w},\boldsymbol{v}} [\tfrac{\xi}{2} \| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} \|_2^2 + \Phi(\boldsymbol{v})] \text{ s.t. } \boldsymbol{v} = \boldsymbol{\Theta}\boldsymbol{w} . \tag{2}$$

where $\boldsymbol{v}$ is the split variable and of the same dimension as $\boldsymbol{\Theta}\boldsymbol{w}$. Then, the SB method can be obtained as [1]

$$\boldsymbol{w}^{(k+1)} = \arg\min_{\boldsymbol{w}} [\tfrac{\xi}{2} \| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{w} \|_2^2 + \tfrac{\xi_c}{2} \| \boldsymbol{\Theta}\boldsymbol{w} - \boldsymbol{v}^{(k)} - \boldsymbol{e}^{(k)} \|_2^2] \tag{3}$$

$$\boldsymbol{v}^{(k+1)} = \arg\min_{\boldsymbol{v}} [\Phi(\boldsymbol{v}) + \tfrac{\xi_c}{2} \| \boldsymbol{\Theta}\boldsymbol{w}^{(k+1)} - \boldsymbol{v} - \boldsymbol{e}^{(k)} \|_2^2] \tag{4}$$

$$\boldsymbol{e}^{(k+1)} = \boldsymbol{e}^{(k)} - \boldsymbol{\Theta}\boldsymbol{w}^{(k+1)} + \boldsymbol{v}^{(k+1)} \tag{5}$$

where $\boldsymbol{e}^{(k)}$ is the estimation error for $\boldsymbol{v}^{(k)}$ at the kth iteration, and constraint parameter ξ_c scales the L_2 regularization. It should be noted that ξ and ξ_c do not need to be independent of each other. The relationship between these parameters and their selection method will be discussed in the next subsection.

The $\boldsymbol{w}$-update in (3) is a standard LS problem, whereas (4) is for the L_1 regularization, which can be solved at extremely fast speed by using soft thresholding [1]:

$$\boldsymbol{v}^{(k+1)} = \text{shrink}(\boldsymbol{\Theta}\boldsymbol{w}^{(k+1)} - \boldsymbol{e}^{(k)}, \tfrac{1}{\xi_c}) \tag{6}$$

where shrink(a, b) = sgn(a)($|a| - b$)$_+$, sgn(a) extracts the sign of a, and $(u)_+$ is equal to u if $u > 0$ and 0 if $u \leq 0$.

For regularized LS problems as in (1), it has been proved that the SB method is convergent even if the updates for the LS problem are inexact [14]. That proof makes it possible to derive a recursive SB method, where an exponential window is applied to the L_1-regularized LS optimization problem (1) as

$$\hat{\boldsymbol{w}}_{\text{RLS}} = \arg\min_{\boldsymbol{w}} \left[\tfrac{\xi}{2} \sum_{i=1}^{n} \lambda_{n-i}(n) \| y_i - \boldsymbol{x}_i^T \boldsymbol{w} \|_2^2 + \Phi(\boldsymbol{\Theta}\boldsymbol{w}) \right] \tag{7}$$

where $\lambda_{n-i}(n) = \lambda\lambda_{n-i-1}(n)$ ($0 < i \leq n$) with $\lambda_0(n) = 1$, and λ is the forgetting factor (FF), which usually satisfies $0 < \lambda \leq 1$. Since we assume $\boldsymbol{w}$ is stationary or varies slowly, we use a fixed FF in this paper, i.e. $\lambda_i(n) = \lambda^i$. In non-stationary environments, λ can be made variable in order to obtain fast tracking [15].

Following the SB method, as in (2) and (3), we can find a solution that minimizes the L_2-regularized recursive LS loss function

$$\boldsymbol{w}_{\text{RLS\_}n}^{(k+1)} = \arg\min_{\boldsymbol{w}} \left[\tfrac{\xi}{2} \sum_{i=1}^{n} \lambda_{n-i}(n) \| y_i - \boldsymbol{x}_i^T \boldsymbol{w} \|_2^2 + \tfrac{\xi_c}{2} \| \boldsymbol{\Theta}\boldsymbol{w} - \boldsymbol{v}^{(k)} - \boldsymbol{e}^{(k)} \|_2^2 \right]. \tag{8}$$

It should be noted that the recursion number n is not necessarily equal to the iteration number k.

The subproblem (8) has the following RLS-optimal solution of $\boldsymbol{w}^{(k+1)}$ at the (k+1)th iteration:

$$\boldsymbol{w}_{\text{RLS\_}n}^{(k+1)} = [\boldsymbol{X}^T(n)\boldsymbol{\Lambda}(n)\boldsymbol{X}(n) + \tfrac{\xi_c}{\xi}\boldsymbol{I}]^{-1}[\boldsymbol{X}^T(n)\boldsymbol{\Lambda}(n)\boldsymbol{y}(n) + \tfrac{\xi_c}{\xi}\boldsymbol{b}^{(k)}] \tag{9}$$

where we have taken advantage of the identity $\boldsymbol{\Theta}^T\boldsymbol{\Theta} = \boldsymbol{I}$, since for most CS problems, the analysis matrix is chosen as a certain orthogonal transformation, such as the discrete Haar wavelet transform. Here, $\boldsymbol{y}(n) = [y_n, y_{n-1} ... y_1]^T$ is a vector stacked from the 1st element to the nth element of $\boldsymbol{y}$, while $\boldsymbol{X}(n) = [\boldsymbol{x}_n, \boldsymbol{x}_{n-1} ... \boldsymbol{x}_1]^T$ is a matrix stacked from the 1st to the nth row of $\boldsymbol{X}$, $\boldsymbol{\Lambda}(n) = \text{diag}([1, \lambda_1(n) ... \lambda_{n-1}(n)])$ is a diagonal matrix of weights, $\boldsymbol{b}^{(k)} = \boldsymbol{\Theta}^T(\boldsymbol{v}^{(k)} + \boldsymbol{e}^{(k)})$, and $\boldsymbol{I}$ is the identity matrix of appropriate size.

The L_2 regularized-RLS problem (9) still remains difficult to fully solve using conventional RLS algorithms. We therfore resort to an efficient QRD structure that has been developed for regularized RLS-like algorithms [4][12]. For this particular application, we firstly use the QRD to update the Cholesky factor $\boldsymbol{R}(n)$ of the covariance matrix $\boldsymbol{R}_{XX}(n) = \boldsymbol{X}^T(n)\boldsymbol{\Lambda}(n)\boldsymbol{X}(n) = \lambda\boldsymbol{R}_{XX}(n-1) + \boldsymbol{x}_n\boldsymbol{x}_n^T$. Let $\overline{\boldsymbol{\Lambda}}(n)\boldsymbol{X}(n) = \boldsymbol{Q}^{(1)}(n)\boldsymbol{R}_0(n)$ be the QRD, where $\overline{\boldsymbol{\Lambda}}(n) = \mathrm{diag}\left[1, \sqrt{\lambda_1(n)}, \cdots, \sqrt{\lambda_{n-1}(n)}\right]$, and $\boldsymbol{R}_0(n) = [\boldsymbol{R}^{(1)}(n); \boldsymbol{0}_{n-L}]$ is an $n \times L$ upper triangular matrix with the $(n-L) \times L$ null matrix appended to $\boldsymbol{R}^{(1)}(n)$. Then, the normal equation for the LS estimate $\tilde{\boldsymbol{w}}(n)$ can be written as $[\boldsymbol{X}^T(n)\boldsymbol{\Lambda}(n)\boldsymbol{X}(n)]\tilde{\boldsymbol{w}}(n) = \boldsymbol{X}^T(n)\boldsymbol{\Lambda}(n)\boldsymbol{y}(n)$. Using the QRD, the above equation becomes $\boldsymbol{R}_0(n)\tilde{\boldsymbol{w}}(n) = [\boldsymbol{Q}^{(1)}(n)]^T\overline{\boldsymbol{\Lambda}}(n)\boldsymbol{y}(n)$. This process can be easily implemented by using the augmented data matrix, as shown by the LS update in part (i) of Table I, i.e. the QRD is first executed for the augmented data vector $[\boldsymbol{x}_n^T, y_n]$. Similarly, the update of the regularization term $\frac{\xi_c}{\xi}\boldsymbol{I}$ is executed successively for each diagonal element using the QRD. Meanwhile, to implement the last term on the right hand side of (9), we successively append $\left[\sqrt{\mu(n)}\boldsymbol{d}_l, \sqrt{\mu(n)}b_l^{(k)}\right]$ to the second QRD at each recursion, where $\boldsymbol{d}_l$ is the lth row of the identity matrix $\boldsymbol{I}$, $b_l^{(k)}$ is the lth element of $\boldsymbol{b}^{(k)}$, and $\mu(n) = (\xi_c/\xi)L$. If the regularization is sequentially applied, then $l = (n \bmod L) + 1$. Thus, $\boldsymbol{w}^{(k+1)}$ can be estimated recursively by using one more observation y_n and one more row of the system matrix $\boldsymbol{x}_n^T$ at each recursion. The arithmetic complexity for updating the QRDs can be reduced to $O(L)$ by using Givens rotation while the back-solving step requires $\frac{1}{2}L(L+1)$ operations. The arithmetic complexity of each competing algorithm is listed in Table II, where it can be seen that for some sparse systems with a sparsity level M, the number of operations $O(ML)$ in the GRLS algorithm may be more than that for the back-solving step. The complexity is further discussed in the simulation results.

Due to the fast convergence of RLS-like algorithms, we can apply the SB update, i.e., the regularization (9), the shrinkage (6), and the update (5), after a few, say T, recursions. The details can be found in part (ii) of Table I. If $T = 1$, the SB update is applied to each recursion, which usually leads to lower mean-square error (MSE) at a cost of slower convergence. As T increases to a moderate number, which means the SB update is applied only when $(n \bmod T) = 0$, the convergence speed could be increased considerably due to more accurate LS estimation. The parameter T, therefore, balances between the convergence rate, which is mainly controlled by the LS update, and the estimation accuracy, which is mainly affected by the SB update. From the simulation results, it can be seen that applying the SB update after a few times, say $T = 4$, the LS update usually leads to satisfactory performance.

B. *Selection of ξ and ξ_c*

The L_1 regularization parameter ξ plays an important role in the performance of the SB-QRRLS algorithm in terms of the steady-state MSE, convergence rate, and tracking capability, while the constraint parameter ξ_c controls the sparsity level of the estimator through the thresholding rule (6). Therefore, we provide in this section a regularization-parameter selection formula for ξ, which leads to an appropriate selection of ξ_c for the thresholding rule.

The regularization parameter ξ is closely related to that of the L_2-regularized RLS-like algorithms developed in [4]. Thus, we derive the selection formula using the detailed performance analysis provided in [4], where the regularization parameter is obtained by balancing between the bias and variance terms of the estimation MSE for white Gaussian inputs. Since L_1 regularization can be approximately implemented by using the L_2 norm through the relationship $|\boldsymbol{w}| = \|\boldsymbol{D}\boldsymbol{w}\|_2^2$ with $\boldsymbol{D}$ the generalized inverse of the estimated coefficient matrix $\operatorname{diag}\left[\sqrt{|\hat{w}_1|}, \cdots, \sqrt{|\hat{w}_L|}\right]$ [4], the analysis can be extended to the L_1-regularized LS algorithm (1) and the regularization parameter ξ can be derived in a similar way. The optimal regularization parameter can be found as

$$\xi_{\text{opt}} = \left[\sigma_x^2 \lambda^{-1} \sqrt{1-\lambda} \sqrt{\tfrac{L}{2}(\sigma_\eta^2 / \sigma_x^2) \|\boldsymbol{w}\|_2^2}\right]^{-1} \tag{10}$$

where σ_x^2 and σ_η^2 are, respectively, the input and system noise variances. In applications such as communication systems, these variances are usually known *a priori* or can be estimated approximately as shown in [15] and [16]. Eq. (10) is derived from the Gaussian input assumption, however, it has been shown [4] that this formula also applies to auto-regressive (AR) processes and other colored inputs, such as speech and music.

The constraint parameter ξ_c in (4) and (6) is closely related to the shrinkage. Although ξ and ξ_c are not necessarily related to each other when deriving the Bregman iteration, we require in this paper that the shrinkage operation in (6) needs to be equivalent to the L_1-regularized LS problem (1). Comparing these two equations and using the thresholding rule in [17], it can be found that ξ_c can be simply chosen as $\xi\sigma_x^2$ for Gaussian distributed inputs with variance σ_x^2. This choice will be used in Section IV for the simulation results.

III. SPCA USING THE MULTIVARIATE SB-QRRLS

SPCA is considered one of the most efficient dimension-reduction techniques. In this section, we derive a multivariate version of the proposed SB-QRRLS algorithm and apply it to the "self-contained" SPCA model [13].

Suppose we are considering the first K principal components (PCs) of the data matrix $\boldsymbol{X}$ of dimension N×L, where N is usually larger than L. Then define $\boldsymbol{W}$ as an L×K matrix, representing the first K PCs. To find a sparse representation of $\boldsymbol{W}$ recursively, we propose the following L_1-regularized exponentially-weighted LS problem

$$\hat{\boldsymbol{W}}(n) = \arg\min_{\boldsymbol{W}} \left[\tfrac{\xi}{2} \sum_{i=1}^{n} \lambda^{n-i} \| \boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{W}^T \boldsymbol{x}_i \|_2^2 + \sum_{j=1}^{K} \| \boldsymbol{w}_j \|_1 \right] \tag{11}$$

where $\boldsymbol{x}_i$ is the ith column of $\boldsymbol{X}^T$, and $\boldsymbol{w}_j$ is the jth column of $\boldsymbol{W}$. To achieve lower estimation MSE, the iteration number k is set equal to n and hence omitted in (11). It can be seen that (11) automatically results in a solution $\boldsymbol{W}$ with orthogonal columns [18]. Therefore, the minimizer under this orthonormal constraint on $\boldsymbol{W}$ is exactly the first K PCs rather than the relaxed restriction [13].

For this multivariate analysis, the Bregman splitting can be applied similarly, yielding

$$(\hat{\boldsymbol{W}}, \hat{\boldsymbol{V}}) = \arg\min_{\boldsymbol{W},\boldsymbol{V}} \left[\tfrac{\xi}{2} \sum_{i=1}^{n} \lambda^{n-i} \| \boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{y}_i \|_2^2 + \sum_{j=1}^{K} | \boldsymbol{v}_j | \right] \quad \text{s.t. } \boldsymbol{V} = \boldsymbol{W} \tag{12}$$

where $\boldsymbol{y}_i = \boldsymbol{W}^T \boldsymbol{x}_i$ and $\boldsymbol{v}_j$ is the jth column of $\boldsymbol{V}$. Then, the LS solution and SB updates for SPCA can be derived as

$$\boldsymbol{W}(n+1) = \arg\min_{\boldsymbol{W}} \left[\tfrac{\xi}{2} \sum_{i=1}^{n} \lambda^{n-i} \| \boldsymbol{x}_i - \boldsymbol{W}\boldsymbol{y}_i \|_2^2 + \tfrac{\xi_c}{2} \| \boldsymbol{W} - \boldsymbol{B}(n) \|_2^2 \right] \tag{13}$$

$$\boldsymbol{v}_j(n+1) = \text{shrink}[\boldsymbol{w}_j(n+1) - \boldsymbol{e}_j(n), \tfrac{1}{\xi_c}] \text{ for } j = 1,2\ldots K \tag{14}$$

$$\boldsymbol{E}(n+1) = \boldsymbol{E}(n) - \boldsymbol{W}(n+1) + \boldsymbol{V}(n+1) \tag{15}$$

where $\boldsymbol{B}(n) = \boldsymbol{V}(n) + \boldsymbol{E}(n)$, and $\boldsymbol{e}_j$ is the jth column of $\boldsymbol{E}$. To solve the LS problem (13), we set the first derivative of the argument to zero to obtain

$$\boldsymbol{W}(n+1) = [\boldsymbol{X}^T(n)\boldsymbol{\Lambda}(n)\boldsymbol{Y}(n) + \tfrac{\xi_c}{\xi}\boldsymbol{B}(n)][\boldsymbol{Y}^T(n)\boldsymbol{\Lambda}(n)\boldsymbol{Y}(n) + \tfrac{\xi_c}{\xi}\boldsymbol{I}]^{-1} \tag{16}$$

where $\boldsymbol{Y}(n) = [\boldsymbol{y}_n, \boldsymbol{y}_{n-1} \ldots \boldsymbol{y}_1]^T = \boldsymbol{X}(n)\boldsymbol{W}(n)$. Eq. (16) can be solved recursively by using QRD. Compared to Table I, QRD needs to remove all the non-zero elements in $\boldsymbol{x}_n$ and $\boldsymbol{b}_l$ in the two updates by using Givens rotation. The detailed implementation of the proposed multivariate SB-QRRLS for SPCA is presented in Table III.

IV. Simulation Results

In this section, we evaluate the proposed SB-QRRLS algorithm for sparse-system estimation and study its application to the SPCA problem. Unless specified otherwise, all results are obtained by averaging over 100 Monte-Carlo simulations.

A. Evaluation of the SB-QRRLS Algorithm

In this experiment, the proposed SB-QRRLS algorithm is evaluated in a sparse-system estimation setting. The unknown system changes from $\mathbf{w}_0$ = [1,–1,1,–1…–1,1] of length L = 25 to a sparse system, where M = 15 coefficients of $\mathbf{w}_0$ are set to zero at the 5000-th sample. The input is simulated by a first-order AR process: $x(n) = 0.95\,x(n-1) + g(n)$, where $g(n)$ is a Gaussian process with zero mean and variance selected so that the power of $\mathbf{x}(n)$ is normalized to 1. The SNR is variously set to 20 dB, 10 dB, and 0 dB. The proposed SB-QRRLS algorithm is compared with the L_1-QRRLS algorithm proposed in [4] and the GRLS algorithm in [10] using predictive least squares (PLS) criteria, which usually outperforms other sparse RLS algorithms in convergence and tracking capability. The results for the fixed-FF RLS algorithm are also presented for reference, where the FF is set to 0.993. The L_1 regularization parameter is set to $\xi = \xi_{\text{opt}}$ per (10), and the constraint parameter $\xi_c = \xi\sigma_x^2 = \xi$ in this case, as discussed at the end of Section II.

The simulated mean-square deviation (MSD) results, which measure the square error of the estimate deviation from its true value $\mathbf{w}_0$, are presented in Fig. 1. It can be seen that for the non-sparse system, the L_1-QRRLS algorithm has a very large MSD compared to the conventional RLS algorithm, due to the bias term introduced by the regularization [4]. The GRLS algorithm converges even slower than the conventional RLS algorithm due to the initial guess of the parameter Δ. The SB-QRRLS algorithms can converge faster to a lower MSD than the other algorithms due to the SB update and soft thresholding. As T increases, e.g. T = 4, 6, the SB-QRRLS algorithm converges even faster at the cost of slightly increased MSD. When the system becomes sparse, the L_1-QRRLS algorithm converges to a lower MSD than the RLS algorithm. The GRLS algorithm has the best performance since it automatically estimates the sparsity of the unknown system at the cost of higher computational complexity. The SB-QRRLS algorithm with T = 4 achieves performance comparable to that of the GRLS algorithm. When the SNR decreases to 0 dB, similar MSD curves can be observed and the improved performance of the SB-QRRLS algorithm is more significant. In this case ($M > \frac{1}{2}L$), the arithmetic complexity of the GRLS algorithm is even more than that for SB-QRRLS, since $O(ML)$ is no longer negligible compared to $\frac{1}{2}L(L+1)$.

To further examine the performance under different conditions, similar simulations are carried out by using a more complicated switching-channel. In this case, the first 5 coefficients of $\boldsymbol{w}_0$ are set 0 and another 15 coefficients (M = 20) are set to 0 at the 5000-th sample while the remaining 5 coefficients change sign as before. The simulation results are presented in Fig. 2, where it can be seen that the proposed SB-QRRLS algorithm still has performance comparable to that of the GRLS algorithm under sparse conditions for both initial convergence and tracking.

B. Effect of the Regularization Parameter ξ

In this experiment, the effect of the regularization parameter ξ is examined. The settings are similar to those in the previous experiment. Simulation results for the steady-state MSD of the SB-QRRLS algorithm are presented in Table IV in different testing environments. In the simulations, we set L = 25, 50; M = 10; the FFs λ = 0.98, 0.99, 0.995; the regularization parameter $\xi = 0.1\,\xi_{\text{opt}}$, ξ_{opt}, $10\,\xi_{\text{opt}}$, where ξ_{opt} is defined in (10); and SNR = 10 dB, 0 dB. From Table IV, it can be seen that the steady-state MSD with $\xi = \xi_{\text{opt}}$ has the smallest value compared to that with either $\xi = 0.1\,\xi_{\text{opt}}$ or $10\,\xi_{\text{opt}}$, which suggests the effectiveness of (10) and the proposed parameter selection method.

C. Application to SPCA

This experiment examines the use of the multivariate SB-QRRLS algorithm for SPCA, as defined in Table III. In this experiment, simulated data is used.

We first generate three mutually independent Gaussian random sequences ($\boldsymbol{g}_1$, $\boldsymbol{g}_2$, $\boldsymbol{\varepsilon}$) of the form $N(0,\sigma^2)$, with zero mean and different variances σ^2, together with their mixture $\boldsymbol{g}_3$:

$$\boldsymbol{g}_1 \sim N(0,200),\quad \boldsymbol{g}_2 \sim N(0,300),\quad \boldsymbol{g}_3 = -0.1\boldsymbol{g}_1 + 0.2\boldsymbol{g}_2 + \boldsymbol{\varepsilon},\ \boldsymbol{\varepsilon} \sim N(0,1) \tag{17}$$

where $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, and $\boldsymbol{\varepsilon}$ are of length 500. The three hidden sequences $\boldsymbol{g}_1$, $\boldsymbol{g}_2$ and, $\boldsymbol{g}_3$ are individually measured by 10 sensors, where the observed variables are

$$\boldsymbol{x}_i = \boldsymbol{g}_1 + \boldsymbol{\varepsilon}_i,\ i = 1,2,3,4, \quad \boldsymbol{x}_i = \boldsymbol{g}_2 + \boldsymbol{\varepsilon}_i,\ i = 5,6,7,8, \quad \boldsymbol{x}_i = \boldsymbol{g}_3 + \boldsymbol{\varepsilon}_i,\ i = 9,10, \tag{18}$$

with $\boldsymbol{\varepsilon}_i \sim N(0,1)$ for all i = 1,2,…,10 mutually independent.

In this case, the variance of the three random sequences $\boldsymbol{g}_1$, $\boldsymbol{g}_2$, and $\boldsymbol{g}_3$ is 200, 300, and 14. Therefore, $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ are much more important than $\boldsymbol{g}_3$, and the first two PCs should be able to recover $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$, showing significant PC values $\hat{\boldsymbol{x}}_i$ from either ($\hat{\boldsymbol{x}}_5$, $\hat{\boldsymbol{x}}_6$, $\hat{\boldsymbol{x}}_7$, $\hat{\boldsymbol{x}}_8$) or ($\hat{\boldsymbol{x}}_1$, $\hat{\boldsymbol{x}}_2$, $\hat{\boldsymbol{x}}_3$, $\hat{\boldsymbol{x}}_4$).

Both conventional SPCA and the proposed SB-based SPCA are carried out by using the prior information that the principal component representations are sparse. In this example, we apply the SB update and the thresholding at each iteration (T = 1) for the SB-based SPCA. The FF is chosen as 0.99. Table V compares the results for the first 3 PCs, which are not averaged by Monte-Carlo simulations in this case. It can be seen that conventional SPCA wrongly includes ($\hat{\boldsymbol{x}}_9$, $\hat{\boldsymbol{x}}_{10}$) in both the first and second most important PCs, due to the correlations of $\boldsymbol{g}_3$ with $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$. The loading of ($\boldsymbol{x}_9$, $\boldsymbol{x}_{10}$) on PC1 is a little bit higher than that on PC2 probably because the correlation between $\boldsymbol{g}_2$ and $\boldsymbol{g}_3$ is higher. On the other hand, the SB-based SPCA correctly identifies the sets of important variables. Thus, multivariate SB-QRRLS delivers ideal sparse representations of the first two PCs, and PC3 correctly turns out to be the null vector, due to the L_1 regularization applied to these sparse loadings.

V. Conclusion

A new adaptive version of the SB method, namely the SB-QRRLS algorithm, has been presented in this short paper, which can be used to recursively solve the L_1-regularized LS problem. An efficient method is developed for selecting the regularization parameter, which controls the sparsity level of the estimator. This algorithm is further extended to a multivariate version and can be applied to SPCA problems. Simulations for sparse estimation illustrate that the proposed SB-QRRLS algorithm compares favorably with conventional algorithms in terms of convergence and tracking performance. Moreover, the multivariate SB-QRRLS shows potential for a variety of applications, such as SPCA for dimension reduction.

Acknowledgment

The work described in this study was fully supported by a grant from the Hong Kong Polytechnic University (The Hong Kong Polytechnic University Postdoctoral Fellowships Scheme, G-YW0L). The first author is indebted to Prof. S. C. Chan for his valuable and stimulating discussions and the inspiration for writing this paper.

References

[1] T. Goldstein and S. Osher, “The split Bregman method for L1-regularized problems,” *SIAM J. Imaging Sciences*, vol. 2, no. 2, pp. 323-343, Apr. 2009.

[2] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, “An iterative regularization method for total variation-based image restoration,” *Multiscale Model. Simul*., vol. 4, no. 2, pp. 460-489, 2005.

[3] P. A. Naylor, J. Cui, and M. Brookes, "Adaptive algorithms for sparse echo cancellation," *Signal Process.*, vol. 86, no. 6, pp. 1182-1192, 2006.

[4] S. C. Chan, Y. J. Chu, Z. G. Zhang, and K. M. Tsui, "A new variable regularized QR decomposition-based recursive least M-estimate algorithm—performance analysis and acoustic applications," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 5, pp. 907-922, May 2013.

[5] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: Where RLS meets the L1-norm," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3436-3447, Jul. 2010.

[6] B. Babadi, N. Kalouptsidis, and V. Tarokh, "SPARLS: The sparse RLS algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4013-4025, Aug. 2010.

[7] N. Kalouptsidis, G. Mileounis, B. Babadi, and V. Tarokh, "Adaptive algorithms for sparse system identification," *Signal Process.*, vol. 91, pp. 1910–1919, 2011.

[8] A. Onose and B. Dumitrescu, "Adaptive matching pursuit using coordinate descent and double residual minimization," *Signal Process.*, vol. 93, no. 11, pp. 3143–3150, 2013.

[9] A. Onose and B. Dumitrescu, "Adaptive randomized coordinate descent for sparse systems: lasso and greedy algorithms," *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 4091-4101, Aug. 2015.

[10] B. Dumitrescu, A. Onose, P. Helin, and I. Tabus, "Greedy sparse RLS," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2194-2207, May 2012.

[11] J. A. Apolinario, *QRD-RLS Adaptive Filtering*, Springer, 2009.

[12] S. C. Chan and Y. J. Chu, "A new state-regularized QRRLS algorithm with a variable forgetting factor," *IEEE Trans. Circuits Syst. II*, vol. 59, no. 3, pp. 183-187, Mar. 2012.

[13] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graph. Stat.*, vol. 15, no. 2, pp. 265-286, Sep. 2006.

[14] N. Hung and J. A. Fessler, "A convergence proof of the split Bregman method for regularized least-squares problems," Available from de.arxiv.org

[15] Y. J. Chu and S. C. Chan, "A new local polynomial modeling-based variable forgetting factor RLS algorithm and its acoustic applications," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 11, pp. 2059-2069, Nov. 2015.

[16] S. H. Leung and C. F. So, "Gradient-based variable forgetting factor RLS algorithm in time-varying environments," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 3141-3150, Aug. 2005.

[17] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *J. Am. Stat. Assoc.*, vol. 96, no. 456, pp. 1348-1360, Dec. 2001.

[18] B. Yang, "Projection approximation subspace tracking," *IEEE Signal Process.*, vol. 43, no. 1, pp. 95-107, Jan. 2004.

TABLES

TABLE I SB-QRRLS ALGORITHM

Initialization: $\boldsymbol{R}(0)=\sqrt{\delta}\boldsymbol{I}$, δ is a small positive constant; $\boldsymbol{u}(0)=\boldsymbol{0}$, $\boldsymbol{w}(0)=\boldsymbol{v}^{(0)}=\boldsymbol{e}^{(0)}=\boldsymbol{0}$ are null vectors. **Recursion:** Given $\boldsymbol{R}(n-1)$, $\boldsymbol{u}(n-1)$, $\boldsymbol{w}(n-1)$, the nth column of $\boldsymbol{X}^T$, $\boldsymbol{x}_n$, the nth element of $\boldsymbol{y}$, y_n, and $\boldsymbol{b}^{(k)}$, we compute at recursion index n: (i). The LS update: $\begin{bmatrix} \boldsymbol{R}^{(1)}(n) & \boldsymbol{u}^{(1)}(n) \\ \boldsymbol{0}^T & c^{(1)}(n) \end{bmatrix} = \boldsymbol{Q}^{(1)}(n) \begin{bmatrix} \sqrt{\lambda}\boldsymbol{R}(n-1) & \sqrt{\lambda}\boldsymbol{u}(n-1) \\ \boldsymbol{x}_n^T & y_n \end{bmatrix}$ (ii). The SB update: **If** (n mod T)==0, $\begin{bmatrix} \boldsymbol{R}(n) & \boldsymbol{u}(n) \\ \boldsymbol{0}^T & c(n) \end{bmatrix} = \boldsymbol{Q}(n) \begin{bmatrix} \boldsymbol{R}^{(1)}(n) & \boldsymbol{u}^{(1)}(n) \\ \sqrt{\mu(n)}\boldsymbol{d}_l & \sqrt{\mu(n)}b_l^{(k)} \end{bmatrix}$ $\boldsymbol{v}^{(k+1)} = \text{shrink}(\boldsymbol{\Theta w}^{(k+1)} - \boldsymbol{e}^{(k)}, \frac{1}{\xi_c})$ $\boldsymbol{e}^{(k+1)} = \boldsymbol{e}^{(k)} - \boldsymbol{\Theta w}^{(k+1)} + \boldsymbol{v}^{(k+1)}$ $k = k+1$ **End if** where $\boldsymbol{Q}^{(1)}(n)$ and $\boldsymbol{Q}(n)$ are calculated by Givens rotation to obtain the left hand side of each equation above and $\mu(n)=(\xi_c/\xi)L$, $\boldsymbol{d}_l$ is the lth row of the identity matrix, and $b_l^{(k)}$ is the lth element of $\boldsymbol{b}^{(k)}$. (iii). $\boldsymbol{w}(n)=\boldsymbol{R}^{-1}(n)\boldsymbol{u}(n)$ (back-substitution).

$\boldsymbol{w}^{(k+1)}$ at the nth recursion index is written as $\boldsymbol{w}(n)$ for short.

TABLE II ARITHMETIC COMPLEXITIES OF VARIOUS ALGORITHMS

	QRRLS	L_1-QRRLS	GRLS	SB-QRRLS
Data update	$5L$	$7L$	$O(ML)+O(L-M)^2$	$(5+\frac{2}{T})L$
Back-solving	$\frac{1}{2}L(L+1)$	$\frac{1}{2}L(L+1)$	$\frac{1}{2}M(M+1)$	$\frac{1}{2}L(L+1)$

M: sparsity level; *L*: filter length.

TABLE III MULTIVARIATE SB-QRRLS FOR SPCA

Initialization:
$\boldsymbol{R}(0)=\sqrt{\delta}\boldsymbol{I}$, δ is a small positive constant;
$\boldsymbol{U}(0)=\mathbf{0}$, $\boldsymbol{W}(0)=\boldsymbol{B}(0)=\boldsymbol{V}(0)=\mathbf{0}$, are null matrices.
Recursion:
Given $\boldsymbol{R}(n-1)$, $\boldsymbol{U}(n-1)$, $\boldsymbol{W}(n-1)$, the *n*th column of $\boldsymbol{Y}(n)^T$, $\boldsymbol{y}(n)$, the *n*th column of $\boldsymbol{X}^T$, $\boldsymbol{x}_n$, and $\boldsymbol{B}(n)$, we compute at recursion index *n*:
(i). The LS update:
$\begin{bmatrix} \boldsymbol{R}^{(1)}(n) & \boldsymbol{U}^{(1)}(n) \\ \mathbf{0}^T & \boldsymbol{c}^{(1)}(n) \end{bmatrix} = \boldsymbol{Q}^{(1)}(n) \begin{bmatrix} \sqrt{\lambda}\boldsymbol{R}(n-1) & \sqrt{\lambda}\boldsymbol{U}(n-1) \\ \boldsymbol{y}^T(n) & \boldsymbol{x}_n \end{bmatrix}$
(ii). The SB update:
$\begin{bmatrix} \boldsymbol{R}(n) & \boldsymbol{U}(n) \\ \mathbf{0}^T & \boldsymbol{c}(n) \end{bmatrix} = \boldsymbol{Q}(n) \begin{bmatrix} \boldsymbol{R}^{(1)}(n) & \boldsymbol{U}^{(1)}(n) \\ \sqrt{\mu(n)}\boldsymbol{d}_l & \sqrt{\mu(n)}\boldsymbol{b}_l(n) \end{bmatrix}$
$\boldsymbol{v}_j(n+1)=\text{shrink}(\boldsymbol{w}_j(n+1)-\boldsymbol{e}_j(n),\frac{1}{\xi_c})$ for $j = 1,2\ldots K$
$\boldsymbol{E}(n+1)=\boldsymbol{E}(n)-\boldsymbol{W}(n+1)+\boldsymbol{V}(n+1)$
where $\boldsymbol{Q}^{(1)}(n)$ and $\boldsymbol{Q}(n)$ are calculated by Givens rotation to obtain the left-hand side of each equation above and $\mu(n) = (\xi_c/\xi)L$, $\boldsymbol{d}_l$ and $\boldsymbol{b}_l(n)$ are, respectively, the *l*th row of the identity matrix and $\boldsymbol{B}(n)$.
(iii). $\boldsymbol{W}(n)=\boldsymbol{U}(n)\boldsymbol{R}^{-1}(n)$ (back-substitution).

TABLE IV COMPARISON OF EXPERIMENTAL STEADY-STATE MSD FOR FIRST-ORDER AR INPUT IN VARIOUS TESTING ENVIRONMENTS

		SNR = 10 dB									SNR = 0 dB								
L	λ	0.98			0.99			0.995			0.98			0.99			0.995		
	ξ	5	50	500	7.1	71	710	10	100	1000	1.7	17	170	2	20	200	3.3	33	330
25	MSD	–4.0	–4.3	–4.1	–7.3	–7.6	–7.5	–10.5	–10.6	–10.5	5.9	5.4	5.8	2.8	2.5	2.9	–0.6	–0.6	–0.4
50	MSD	–0.4	–0.4	–0.3	–3.8	–3.8	–3.6	–6.8	–6.8	–6.5	9.4	9.4	9.7	6.2	5.9	6.2	2.9	2.9	3.2

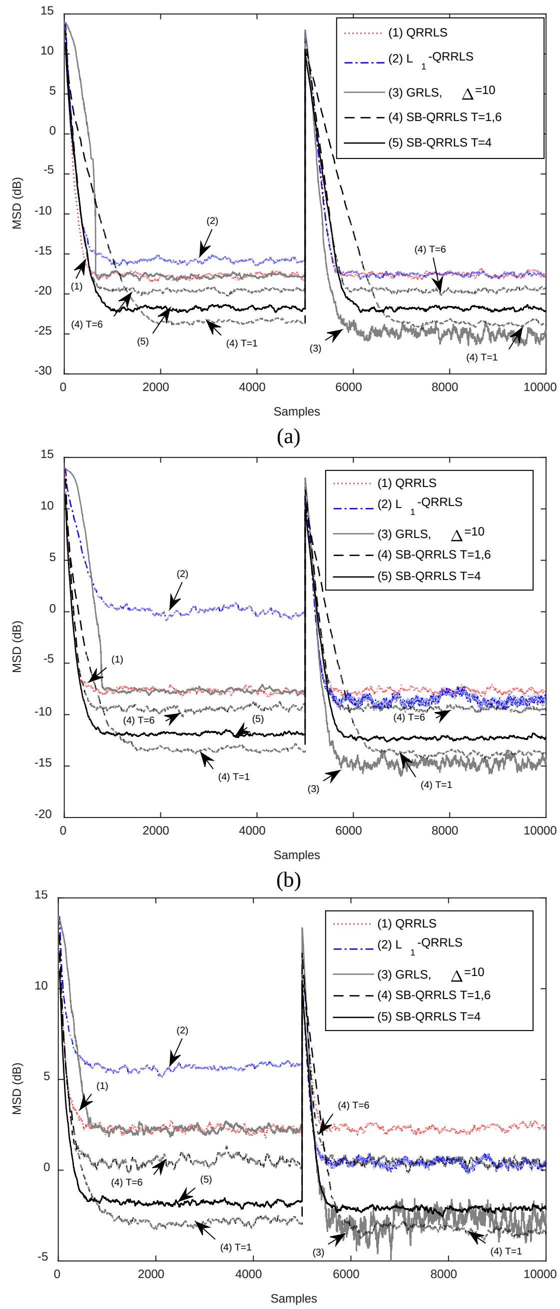
TABLE V RESULTS OF PCA AND SB-PCA ANALYSIS

	PCA			SB-PCA		
	PC1	PC2	PC3	PC1	PC2	PC3
$\hat{\boldsymbol{x}}_1$	–0.189	0.390	0.118	0.000	0.498	0.000
$\hat{\boldsymbol{x}}_2$	–0.203	0.364	0.162	0.002	0.500	0.000
$\hat{\boldsymbol{x}}_3$	–0.179	0.403	0.094	–0.001	0.495	0.000
$\hat{\boldsymbol{x}}_4$	–0.191	0.400	0.108	0.002	0.502	0.000
$\hat{\boldsymbol{x}}_5$	0.354	0.158	0.320	–0.500	–0.000	0.000
$\hat{\boldsymbol{x}}_6$	0.363	0.181	0.288	–0.495	0.002	0.000
$\hat{\boldsymbol{x}}_7$	0.346	0.157	0.328	–0.494	0.005	0.000
$\hat{\boldsymbol{x}}_8$	0.353	0.171	0.308	–0.495	0.005	0.000
$\hat{\boldsymbol{x}}_9$	0.289	0.185	–0.537	–0.062	–0.041	0.000
$\hat{\boldsymbol{x}}_{10}$	0.282	0.171	–0.518	–0.057	–0.040	0.000

FIGURE CAPTION

Fig. 1. Comparison of MSD curves, where the channel changes from non-sparse during convergence to sparse during tracking, for various algorithms with 1st-order AR input and $L = L_0 = 25$. (a) SNR = 20 dB, (b) SNR = 10dB, (c) SNR = 0dB.

Fig. 2. Comparison of MSD curves, where the channel is sparse during both convergence and tracking, for various algorithms with 1st-order AR input and $L = L_0 = 25$. (a) SNR = 20 dB, (b) SNR = 10 dB, (c) SNR = 0 dB.



(c)

Fig. 1. Comparison of MSD curves, where the channel changes from non-sparse during convergence to sparse during tracking, for various algorithms with 1st-order AR input and $L = L_0 = 25$. (a) SNR = 20 dB, (b) SNR = 10dB, (c) SNR = 0dB.

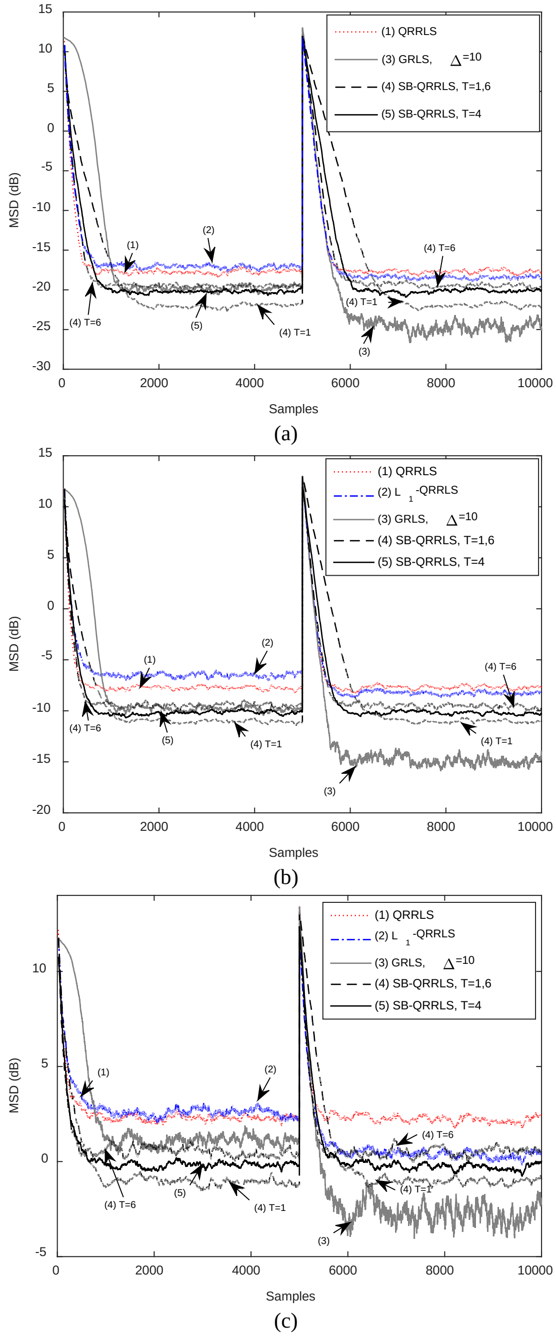


Fig. 2. Comparison of MSD curves, where the channel is sparse during both convergence and tracking, for various algorithms with 1st-order AR input and $L = L_0 = 25$. (a) SNR = 20 dB, (b) SNR = 10 dB, (c) SNR = 0 dB.