SPARSE CONSTRAINT AFFINE PROJECTION ALGORITHM WITH PARALLEL IMPLEMENTATION AND APPLICATION IN COMPRESSIVE SENSING

Dong Yin† †H. C. So*† Yuantao Gu††

†Tsinghua National Laboratory for Information Science and Technology, Department of Electronic Engineering, City University of Hong Kong, Hong Kong, CHINA
* Department of Electronic Engineering, City University of Hong Kong, Hong Kong, CHINA

ABSTRACT

Based on affine projection algorithm (APA) in adaptive filtering and the technique of parallel computing, we propose a novel algorithm called \(\ell_0\)-APA with its parallel implementation for sparse system identification and sparse signal recovery. For sparse system identification, parallel \(\ell_0\)-APA can serve as an effective approach for practical hardware implementation, since it lowers the requirement on the processors’ clock speed. For sparse signal recovery, it can significantly reduce the convergence time. Prior algorithms such as \(\ell_0\)-LMS and \(\ell_0\)-ZAP can be seen as special cases of \(\ell_0\)-APA. Finally, the performance of the proposed algorithm is analyzed and verified by numerical experiments.

Index Terms— Affine projection algorithm, parallel implementation, steady-state behavior, transient behavior, compressive sensing

1. INTRODUCTION

Since the unknown systems in the real world are often sparse [2, 3], there has long been considerable interest in solving sparse system identification problems [4, 5]. Sparsity indicates a small proportion of nonzero coefficients in a long unknown impulse response. Without the exploration of sparsity, traditional adaptive filtering algorithms such as least mean square (LMS), recursive least squares (RLS) and affine projection algorithm (APA) never show further effectiveness in sparse system identification. Consequently, some algorithms utilizing the prior knowledge of sparsity have been proposed in recent years. M-Max Normalized LMS (MMax-NLMS) [6], Gradient Projection (GP) [7], Sequential Partial Update LMS (S-LMS) [8] and Proportionate NLMS (PNLMS) [2, 9] are several examples. Established in the past few years, a novel branch of sparse signal processing, compressive sensing (CS) [10, 11], gives theoretical guarantee of successful reconstruction from highly incomplete measurements. Borrowing the sparseness penalty from some recovery algorithms in CS, several new methods for sparse system identification such as ZA-LMS [12] and \(\ell_0\)-ZAP [13, 14] have been developed recently. Besides the successful application in system identification, \(\ell_0\)-LMS may also serve as an effective reconstruction algorithm for CS. The relationship between the adaptive filtering framework and compressed signal recovery is addressed in [14]. By using a zero-point attraction as the sparse penalty, the above adaptive algorithms have behaved pretty well in both sparse system identification and sparse signal recovery. It has been numerically verified [14] and theoretically proved [15] that the overall performance is enhanced in both steady-state mean-square deviation (MSD) and probability of successful reconstruction, especially in the practical noisy scenarios. Very recently, there are several algorithms including TD-ZA-LMS [16] and non-uniform norm constraint LMS [17] proposed based on similar ideas.

However, the drawbacks of \(\ell_0\)-LMS and other similar algorithms are obvious. They are not practical approaches for hardware implementation due to their sample-by-sample iterative process. The processor has to finish updating the filter tap-weights before next sample comes in. This often results in expensive or impractical high requirements on the processors’ clock speed. Designing algorithms with parallel implementation has become an effective way to solve such kind of problems [18, 19]. Based on the advancement of integrated circuits, parallel computing has been popular especially via digital processors such as FPGA and GPU. Parallel algorithms would make it possible to finish a task with lower clock frequency than their serial counterparts, and at the same time no prolonged period of time is needed due to the efficient parallel structures.

Affine projection algorithm (APA) [20] is an effective approach in adaptive filtering particularly for applications such as acoustic echo cancellation [21], active noise control [22] and distributed estimation [23]. Compared to other filtering algorithms such as LMS and normalized LMS, APA has shown better behavior in the scenarios of colored input signal [24]. Several types of fast implementation such as fast affine projection (FAP) [24] and block exact fast affine projection (BEFAP) [25], have been proposed to reduce computational complexity. On the other hand, the steady-state performance and convergence behavior of APA have been extensively studied in [26, 27].

In order to solve the difficulty in hardware implementation caused by the serial structure of available sparsity included adaptive algorithms, we propose a new sparse constraint algorithm based on APA called \(\ell_0\)-APA with its practical parallel implementation for sparse system identification and sparse signal recovery. We further demonstrate that the parallel implementation is more advantageous in the later than in the former. Because the reconstruction in CS is usually high-dimensional and very expensive in time complexity, parallel computing can significantly reduce the time to attain the desired results, especially in online scenarios. In addition, prior algorithms such as \(\ell_0\)-LMS and \(\ell_0\)-ZAP [14] can be regarded as special cases of the proposed algorithm. Performance analysis on \(\ell_0\)-APA is also conducted and justified by experiments in this work.

†More details of this work can be found in [1]. This work was partially supported by the National Program on Key Basic Research Project (973 Program 2013CB325201) and the National Natural Science Foundation of China (NSFC 61371137). The corresponding author of this paper is Yuantao Gu (gyt@tsinghua.edu.cn).
2. $\ell_0$-APA and its Parallel Implementation

We denote input signal, desired signal, unknown coefficients, and adaptive tap-weights by, respectively, $x_n, d_n, s \in \mathbb{R}^N$, and $w_n \in \mathbb{R}^N$, where $n$ and $N$ are the time instant and filter length. One has $d_n = x_n^T s + e_n$, where $x_n = [x_n, x_{n-1}, \ldots, x_{n-N+1}]^T$ and $e_n$ denotes input vector and measurement noise, respectively. We follow the assumption of Block APA [25] that the adaptive algorithm works once after $M$ new samples are collected. Therefore, the input of $\ell_0$-APA is a matrix of data

$$X_k = [x_{k,M}, x_{k,M-1}, \ldots, x_{k(k-1)+1, M+1}]^T$$

and a vector of $d_k = [d_{k, M}, d_{k, M-1}, \ldots, d_{k(k-1)+1, M+1}]^T$ with relation $d_k = x_k^T s + e_{k, M}$, where $v_{k, M} = [v_{k, M}, v_{k, M-1}, \ldots, v_{k(k-1)+1, M+1}]^T$ denotes the noise vector and $k$ is the iteration instant.

Borrowing the idea of zero-point attraction in $\ell_0$-LMS, we may modify Block APA [25] to explore the sparsity of unknown systems. The proposed $\ell_0$-APA may be summarized into a two-step procedure, i.e., a zero-point attraction on the tap-weights is conducted and followed by a projection onto the solution space of $d_k = X_k s$. The procedure of $\ell_0$-APA is given in Table 1. In this algorithm, $g(w_n) = [g(w_{n, 1}), g(w_{n, 2}), \ldots, g(w_{n, N})]^T$ denotes the zero-point attraction, which is commonly used in [14, 15], where $g(t)$ is defined as follows,

$$g(t) = \begin{cases} -\alpha^2 t + \text{asgn}(t), & 0 < |t| \leq 1/\alpha; \\ 0, & \text{elsewhere}. \end{cases} \quad (1)$$

One may readily find that $\ell_0$-APA is a generalization of $\ell_0$-LMS. In fact, $\ell_0$-APA degenerates to the latter in the situations with $M = 1$.

Next we consider the parallel implementation of the proposed sparse constraint APA. In the parallel scenario, we assume that there are $Q$ parallel processors that work separately and independently at the same time. Therefore, $Q$ adjacent input matrices

$$X_k = [X_{kQ}, X_{k(Q-1)}, \ldots, X_{k(Q-1)+1}]^T$$

and desired vectors $d_k = [d_{kQ}, d_{k(Q-1)}, \ldots, d_{k(Q-1)+1}]^T$ may be processed simultaneously, where $l$ denotes the parallel iteration number. Therefore, after one step of zero-point attraction, the projections to respective solution spaces of $\{d_{kQ = l} = X_{kQ} s, s \}$ are implemented on separated processors, while the increments of all these projections are summed up at the end of the iteration to finally update the tap-weights. The detailed procedures of Parallel $\ell_0$-APA are in Table 2, where $\Delta x$ denotes the increment of the projection in $(i + 1)$th processor at $l$th parallel iteration.

As stated in Section 1, the disadvantages of serial algorithms including $\ell_0$-LMS and $\ell_0$-APA are obvious. The single processor has to finish all tasks, hence it results in a high requirement on the processor’s clock frequency. In the parallel computing scenario, $Q$ processors may work together to update the tap-weights. Therefore, the clock frequency of parallel implementation may reduce to $1/Q$ of the serial one. Actually the parallel algorithm is a trade-off between the area and the clock frequency of the processors. We may notice that it is inevitable for the parallel algorithm to have time delay as it updates the tap-weights less frequently. However, since we can still use the latest tap-weights to filter the inputs, the time delay for filtered output $y_n$ and estimation error $e_n$ is not a significant sacrifice.

3. Performance Analysis

Since parallel $\ell_0$-APA is equivalent to $\ell_0$-APA mathematically when we choose $Q = 1$, we only consider the theoretical analysis of parallel $\ell_0$-APA in this section.

3.1. Assumptions

As is stated in [15], the unknown coefficients could be classified into three groups: large, small, and zero. The sets of the indices of these groups are denoted by $C_L, C_S$, and $C_0$, respectively. We assume that entries of $X_k$ and $v_k$ follow i.i.d. Gaussian distribution $\mathcal{N}(0, P_n)$.
and $\mathcal{N}(0, P_s)$ respectively. The assumptions in [15] are adopted in this work in order to simplify the analysis. In addition, we add a new assumption that $(X, X^T)^{-1} \approx 1/N P_n$.

3.2. Steady-state Performance

For simplicity we take $\mu = 1$ in the following sections. Denote the deviation of $w$ by $h_s$, i.e., $h_s = w_{s,0}$. And we use $\Sigma_{1,\infty} = \sum_{i \in C_1} E\{h_{1,s,0}^2\}$ and $\Sigma_{2,\infty} = \sum_{i \in C_2} E\{h_{2,s,0}^2\}$ to denote expectation of the sum of the squared deviations of the first two groups. Here $E$ denotes the expectation operator. For the zero coefficient group, $w = (E\{h_{0,s,0}^2\})^{1/2} (i \in C_0)$ is adopted to represent the deviation of the tap-weights. Another three notations we will utilize are $D_n = E\{|h_i|^2\}$, $P_n = E\{|h_i|^2 g(w_s)|^2\}$, and $G_n = E\{|h_i|^2 g(w_s)|^2\}$. Armed with these assumptions and notations, the following theorem on the steady-state behavior of parallel $\ell_0$-APA is derived.

**Theorem 1** Let

$$S_\infty = \begin{bmatrix} \Sigma_{L,\infty} & \Sigma_{S,\infty} & D_\infty & P_\infty & G_\infty \end{bmatrix},$$

Then $S_\infty$ and $w$ are approximately determined by the following linear and quadratic equations

$$D(w) = BS_\infty,$$\hspace{1cm}(2)

$$d_2 w^2 + d_1 w + (-D_\infty - 2\gamma P_\infty + \gamma^2 G_\infty) + f = 0,$$\hspace{1cm}(3)

where the matrix $B$, vector $D(w)$, scalars $d_1, d_2, \gamma$, and $f$ are given in subsection 7.1.

The proof of this theorem is in section 7. Note that $D_n$ is the mean-square deviation (MSD) of parallel $\ell_0$-APA after the $n$th update and consequently $D_\infty$ is the steady-state MSD. Since the highest degree of the polynomials of $w$ appeared in the equations is 2, these equations can be solved analytically. Then we can get the steady-state MSD.

Due to the limited space this paper, the analysis of transient behavior is omitted. More details can be found in [1].

4. APPLICATION IN COMPRESSIVE SENSING

In CS, the problem of reconstructing a sparse signal $x \in \mathbb{R}^N$ from the highly incomplete measurements $y = Ax + v$, where $A \in \mathbb{R}^{M \times N}$ and $M \ll N$ is considered. The relationship between adaptive filtering framework and compressive sensing has been addressed by [14].

To make reasonable comparison, we divide $A$ into $Q$ submatrices $A_0, A_1, \ldots, A_{Q-1}$ of size $M/Q \times N$ and divide $y$ and $v$ into $y_i$ and $v_i$ ($0 \leq i \leq Q - 1$) correspondingly. Then there is $y_i = A_i z + v_i$. For $\ell_0$-APA, we choose $X_i = A_{	ext{mode}}(h_{0,i})$, which means the $Q$ sub-matrices are put into the adaptive filtering framework circularly. In the scenarios of $Q = M$ and $Q = 1$, $\ell_0$-APA becomes the $\ell_0$-LMS and $\ell_0$-ZAP respectively. Therefore, $\ell_0$-APA is also a general framework for prior algorithms for CS. For parallel $\ell_0$-APA, we choose $X_i = A$ for all $i \geq 0$, which means the $Q$ sub-matrices are processed in parallel in each iteration. In this way, the serial processor and each independent processor of the parallel algorithm share the same computational complexity since they all deal with an $M/Q \times N$ sub-matrix. When the two processors have the same clock frequency, the parallel algorithm will converge much faster because $Q$ projections are conducted at the same time. This will be illustrated by experiments in section 5.

5. NUMERICAL EXPERIMENTS

Experiments are conducted to justify the presented results.\(^1\)

Figure 1 illustrates the accuracy of the steady-state performance analysis in Theorem 1. In this experiment, we investigate how the steady-state MSD changes with parameters $\gamma$, $\alpha$, and $Q$ and compare the results by the simulations and Theorem 1. Note that in the experiment with $Q$, we keep the product of $Q$ and $M$ the same. This can illustrate the relationship between the steady-state MSD and the number of parallel processors when the update frequency is kept the same. From the notations in section 7.1, we know that the steady state MSD is a function of the product of $Q$ and $M$, and the experiment verifies this since the steady-state MSD does not change significantly with $Q$.

Figure 2 shows that for CS, the probability of exact reconstruction of parallel $\ell_0$-APA is approximately the same as prior algorithms such as $\ell_0$-LMS and $\ell_0$-ZAP. This figure implies that we may need less measurements to recover a sparse signal using the adaptive filtering framework than referred methods.

Figure 3 demonstrates that for CS, to achieve the same steady-state MSD, the parallel $\ell_0$-APA has the highest convergence speed among $\ell_0$-LMS, $\ell_0$-ZAP, and $\ell_0$-APA. This advantage naturally comes from the parallel structure.

6. CONCLUSIONS

The proposed algorithm $\ell_0$-APA is a general framework for prior algorithms and its parallel implementation is an effective approach to sparse system identification and CS. For sparse system identification, parallel $\ell_0$-APA lowers the requirements on the clock speed of

\(^1\)The code for these experiments is available at
http://gu.ee.tsinghua.edu.cn/publications#yd1
In this section we give the detailed expressions of the processor, and for CS, the parallel algorithm enhances the convergence speed. The performance analysis in this work is also in good accordance with the simulation results.

7. APPENDIX

7.1. Notations

In this section we give the detailed expressions of $B$, $D(w)$, $d_1$, $d_2$, $e$, and $f$. First denote the cardinality of $C_L$, $C_S$, and $C_o$ by $N_L$, $N_S$, and $N_O$. Then we have

$$d_1 = -2\sqrt{2/\pi} \alpha C_1 (1 + \gamma \alpha^2), \quad d_2 = C_1 C_3 - 1,$$

$$e = MQ/N^2, \quad f = e P_o / P_a + C_1 \gamma^2 \alpha^2,$$

$$B = \begin{bmatrix}
    C_1 - 1 & 0 & e N_L & -2 \gamma \alpha N_L \\
    C_1 C_4 - 1 & e N_S & -2 \gamma \alpha N_S & \gamma^2 \alpha^2 \\
    0 & 1 & 1 & 0 \\
    0 & \alpha^2 & 0 & 1 \\
    0 & 0 & 0 & 1
\end{bmatrix},$$

$$D(w) = \begin{bmatrix}
    -e P_o N_L / P_a \\
    C_1 (C_1 C_6 - \gamma^2 C_3) - e P_o N_S / P_a \\
    -N_0\alpha^2 \\
    C_5 - 2 \alpha^2 C_6 + N_0 \alpha^2 (\gamma^2 \alpha^2 - 2/\pi \alpha^2 + 1)
\end{bmatrix},$$

where $C_1 = 1 - M Q / N + M Q (M Q + 1) / N^2$, $C_2 = (1 + \gamma \alpha^2) (1 - M Q / N)$, $C_3 = 1 + 2 \alpha^2 \gamma + \gamma^2 \alpha^2$, $C_4 = 2 \gamma (1 + \gamma \alpha^2)$, $C_5 = \|g(s)\|^2_2$, and $C_6 = \gamma (1 - MQ/N) C_5 / (C_2 - 1)$.

7.2. Proof of Theorem 1

Proof: The update principle given by parallel $E_o$-APA is

$$w_{t+1} = w_t + \sum_{i=0}^{Q-1} X_{t+1}^T (X_{t+1} X_{t+1}^T)^{-1} (d_{t+1} - X_{t+1} w_t).$$

Considering $w_{t+1} = h_{t+1} + s$, $w_{t+1} = w_t - \gamma g(w_t)$, $d_{t+1} = X_{t+1} + \gamma v_{t+1}$, and using the assumption that $(X_{t+1} X_{t+1}^T)^{-1} \approx I / N P_a$, we know that the following equation holds approximately

$$h_{t+1} = h_t - \gamma g(w_t) + \frac{1}{N P_a} \sum_{i=0}^{Q-1} X_{t+1}^T (-X_{t+1} (h_t - \gamma g(w_t)) + v_{t+1}) + \gamma g(w_t) + v_{t+1}.$$

By taking expectation on $h_{t+1}^T$ and after a series of derivation, we get

$$E\{h_{t+1}^T h_{t+1}\} = \left(1 - \frac{2 M Q}{N} + \frac{(Q - 1)/2 M^2}{N^2}\right) E_{t,0} + \frac{Q}{N^2 P_a} (E_{t,1} - \gamma E_{t,2} + \gamma^2 E_{t,3}) + \frac{M Q P_a}{N^2 P_a},$$

where

$$E_{t,0} = E\{h_t h_t^T\} - \gamma E\{g(w_t) h_t^T\} + \gamma g(w_t) h_t^T,$$

$$E_{t,1} = E\{X_{t+1} X_{t+1}^T h_t h_t^T X_{t+1}^T X_{t+1}\},$$

$$E_{t,2} = E\{X_{t+1}^T X_{t+1} h_t g(w_t) h_t^T X_{t+1}^T X_{t+1}\},$$

$$E_{t,3} = E\{X_{t+1}^T X_{t+1} g(w_t) g(w_t) h_t^T X_{t+1}^T X_{t+1}\}.$$

Since the columns of $X_{t+1}$ are $x_{t+1} m - 1, 0 \leq i \leq M - 1$, there is

$$E_{t,1} = E\left\{\sum_{i=0}^{M-1} x_{t+1} x_{t+1}^T h_t h_t^T \sum_{i=0}^{M-1} x_{t+1} x_{t+1}^T\right\} = M E\{x_{t+1} x_{t+1}^T h_t h_t^T x_{t+1}^T x_{t+1}\} + M (M - 1) \sum_{i=0}^{M-1} P_a^2 E\{h_t h_t^T\}$$

$$= M \left[P_a^2 \text{tr} (P_a E\{h_t h_t^T\}) I + M P_a E\{h_t h_t^T\}\right] + M (M - 1) \sum_{i=0}^{M-1} P_a^2 E\{h_t h_t^T\}$$

$$= M P_a^2 \left[D_{t-1} I + (M + 1) E\{h_t h_t^T\}\right],$$

where $\text{tr} (\cdot)$ denotes the trace of matrices. Similar results can be derived for $E_{t,2}$ and $E_{t,3}$. By substituting the expressions of $E_{t,1}$, $E_{t,2}$ and $E_{t,3}$ into $E\{h_{t+1} h_{t+1}^T\}$ and taking out the ith element on the diagonal of $E\{h_{t+1} h_{t+1}^T\}$, we get

$$E\{h_{t+1,i}^2\} = C_1 (E\{h_{t,i}^2\} - 2 \gamma E\{h_{t,i}^2 \gamma (w_{t,i})\}) + \gamma^2 E\{g(w_{t,i})^2\}$$

$$+ \frac{M Q P_a}{N^2} (D_{t-1,i} - 2 P_{a-1} + 2 \gamma C_{t-1,i}) + \frac{M Q P_a}{N^2 P_a},$$

where $C_1$ is defined in subsection 7.1. By analyzing the situations when $i \in C_o, i \in C_S$, and $i \in C_o$, respectively with the method in [15], we arrive at the conclusion. \qed
8. REFERENCES


