A STOCHASTIC 3MG ALGORITHM WITH APPLICATION TO 2D FILTER IDENTIFICATION

Emilie Chouzenoux[1]*, Jean-Christophe Pesquet*[1]*, and Anisia Florescu*[2]

[1] Université Paris-Est, LIGM, UMR CNRS 8049, Champs sur Marne, France

[2] Dunărea de Jos University, Electronics and Telecommunications Dept., Galaţi, România

ABSTRACT

Stochastic optimization plays an important role in solving many problems encountered in machine learning or adaptive processing. In this context, the second-order statistics of the data are often unknown a priori or their direct computation is too intensive, and they have to be estimated on-line from the related signals. In the context of batch optimization of an objective function being the sum of a data fidelity term and a penalization (e.g. a sparsity promoting function), Majorize-Minimize (MM) subspace methods have recently attracted much interest since they are fast, highly flexible and effective in ensuring convergence. The goal of this paper is to show how these methods can be successfully extended to the case when the cost function is replaced by a sequence of stochastic approximations of it. Simulation results illustrate the good practical performance of the proposed MM Memory Gradient (3MG) algorithm when applied to 2D filter identification.

Index Terms— stochastic approximation, optimization, subspace algorithms, memory gradient methods, descent methods, recursive algorithms, majorization-minimization, filter identification, Newton method, sparsity, machine learning, adaptive filtering.

1. INTRODUCTION

We consider a sequence of random variables $(\mathbf{X}_n, \mathbf{y}_n)_{n\geqslant 1}$ taking their values in $\mathbb{R}^{N\times Q} \times \mathbb{R}^Q$, defined on a probability space $(\Omega, \mathcal{F}, \mathsf{P})$. Our objective is to solve the following minimization problem:

$$\underset{\boldsymbol{h}\in\mathbb{R}^N}{\text{minimize}} \;\; F(\boldsymbol{h}) \tag{1}$$

where

$$(\forall \boldsymbol{h} \in \mathbb{R}^N) \quad F(\boldsymbol{h}) = \frac{1}{2}\mathsf{E}\big(\|\mathbf{y}_n - \mathbf{X}_n^\top \boldsymbol{h}\|^2\big) + \Psi(\boldsymbol{h}). \tag{2}$$

Throughout this paper, $\mathsf{E}(\cdot)$ denotes the mathematical expectation, $\|\cdot\|$ is the Euclidean norm, and Ψ is a function from $\mathbb{R}^N$ to $\mathbb{R}$, which plays the role of a regularization function. In particular, this function may be useful to incorporate some prior knowledge about $\boldsymbol{h}$, e.g. some sparsity requirement, possibly in some transformed domain. We assume here that the following wide-sense stationarity properties hold:

$$(\forall n \in \mathbb{N}^*) \qquad \mathsf{E}(\|\mathbf{y}_n\|^2) = \varrho \tag{3}$$

$$\mathsf{E}(\mathbf{X}_n \mathbf{y}_n) = \boldsymbol{r} \tag{4}$$

$$\mathsf{E}(\mathbf{X}_n \mathbf{X}_n^\top) = \boldsymbol{R} \tag{5}$$

where $\varrho \in]0, +\infty[$, $\boldsymbol{r} \in \mathbb{R}^N$, and $\boldsymbol{R} \in \mathbb{R}^{N\times N}$ is a symmetric positive semi-definite matrix.

Many optimization algorithms can be devised to solve Problem (1) depending on the assumptions made on Ψ [1, 2, 3]. In this work, we will be interested in Majorize-Minimize (MM) subspace algorithms [4]. These approaches proceed by building at each iteration a simple majorant (e.g. a quadratic majorant) of the cost-function, which is minimized in a subspace of low dimension. This subspace is often restricted to the gradient computed at the current iterate and to a memory part (e.g. the difference between the current iterate and a previous one). In a number of recent works [5, 6, 7], these algorithms are shown to provide fast numerical solutions to optimization problems involving smooth functions, in particular in the case of large-scale problems. Note that, although our approach will assume that Ψ is a differentiable function, it has been shown that tight approximations of non-smooth penalizations such as ℓ_1 (resp. ℓ_0) functions, namely $\ell_2 - \ell_1$ (resp. $\ell_2 - \ell_0$) functions, can be employed and are often quite effective in practice [6, 7]. Another advantage of the class of optimization methods under investigation is that their convergence can be established under some technical assumptions, even in the case when Ψ is a nonconvex function (see [6] for more details).

One of the difficulties encountered in machine learning or adaptive processing is that Problem (1) cannot be directly solved since the second-order statistical moments ϱ, $\boldsymbol{r}$ and $\boldsymbol{R}$ are often unknown a priori or their direct computation is too intensive, and they have thus to be estimated on-line from the related time series. In the simple case when $\Psi = 0$, the classical Recursive Least Squares (RLS) algorithm can be used for this purpose [8]. When Ψ is nonzero, stochastic approximation algorithms have been developed such as the celebrated stochastic gradient descent (SGD) algorithm [9]. This algorithm has been at the origin of a tremendous amount of works. It is known to be robust and easy to implement, but its con-

vergence speed may be relatively slow. Various extensions of this algorithm have been developed to alleviate this problem [10, 11], to make it adaptive, or to improve its performance when estimating sparse vectors [12, 13]. When $\Psi \propto \|\cdot\|_1$, an on-line variant of the RLS algorithm was designed in [14] which relies on a coordinate descent approach.

Designing Majorize-Minimize optimization algorithms in a stochastic context constitutes a challenging task since most of the existing works have been focused on batch optimization procedures, and the related convergence proofs usually rely on deterministic tools. We can however mention a few recent works [15, 16] where stochastic MM algorithms are investigated for general loss functions under an independence assumption on the involved random variables, but without introducing any search subspace. Works which are more closely related to ours are those based on Newton or quasi-Newton stochastic algorithms [17, 18, 19, 20], in particular the approaches in [19, 20] provide extensions of BFGS algorithm, but proving the convergence of these algorithms requires some specific assumptions. Like BFGS algorithm, MM subspace methods use a memory of previous estimates so as to accelerate the convergence.

In Section 2, we show how Problem (1) can be reformulated in a learning context. The MM strategy which is proposed in this work is described in Section 3.1. In Section 3.2, we give the form of the resulting recursive algorithm and, in Section 3.3, we evaluate its computational complexity. In Section 4, we show the good performance of the proposed stochastic Majorize-Minimize Memory Gradient (3MG) algorithm for solving a two-dimensional filter identification problem. Some conclusions are drawn in Section 5.

2. PROBLEM FORMULATION

In a learning context, function F can be replaced by a sequence $(\mathrm{F}_n)_{n\geqslant 1}$ of stochastic approximations of it, which are defined as: for every $n \in \mathbb{N}^*$,

$$(\forall \boldsymbol{h} \in \mathbb{R}^N) \qquad \mathrm{F}_n(\boldsymbol{h}) = \frac{1}{2n}\sum_{k=1}^{n} \|\mathbf{y}_k - \mathbf{X}_k^\top \boldsymbol{h}\|^2 + \Psi(\boldsymbol{h}) = \frac{1}{2}\rho_n - \mathbf{r}_n^\top \boldsymbol{h} + \frac{1}{2}\boldsymbol{h}^\top \mathbf{R}_n \boldsymbol{h} + \Psi(\boldsymbol{h}) \tag{6}$$

where ρ_n, $\mathbf{r}_n$, and $\mathbf{R}_n$ are the following classical sample estimates of ϱ, $\boldsymbol{r}$, and $\boldsymbol{R}$:

$$\rho_n = \frac{1}{n}\sum_{k=1}^{n} \|\mathbf{y}_k\|^2 \tag{7}$$

$$\mathbf{r}_n = \frac{1}{n}\sum_{k=1}^{n} \mathbf{X}_k \mathbf{y}_k \tag{8}$$

$$\mathbf{R}_n = \frac{1}{n}\sum_{k=1}^{n} \mathbf{X}_k \mathbf{X}_k^\top. \tag{9}$$

Our objective in the next section will be to propose an efficient method for minimizing F_n, for every $n \in \mathbb{N}^*$.

3. PROPOSED METHOD

3.1. Majorization property

At each iteration $n \in \mathbb{N}^*$, we propose to replace F_n by a surrogate function $\Theta_n(\cdot, \mathbf{h}_n)$ based on the current estimate $\mathbf{h}_n$ (computed at the previous iteration). More precisely, a tangent majorant function is chosen such that

$$(\forall \boldsymbol{h} \in \mathbb{R}^N) \qquad \mathrm{F}_n(\boldsymbol{h}) \leqslant \Theta_n(\boldsymbol{h}, \mathbf{h}_n) \tag{10}$$

$$\mathrm{F}_n(\mathbf{h}_n) = \Theta_n(\mathbf{h}_n, \mathbf{h}_n). \tag{11}$$

For the so-defined MM strategy to be worthwhile, the surrogate function has to be built in such a way that its minimization is simple. For this purpose, we will assume that the regularization function Ψ has the following form:

$$(\forall \boldsymbol{h} \in \mathbb{R}^N) \quad \Psi(\boldsymbol{h}) = \frac{1}{2}\boldsymbol{h}^\top \boldsymbol{V}_0 \boldsymbol{h} - \boldsymbol{v}_0^\top \boldsymbol{h} + \sum_{s=1}^{S} \psi_s(\|\boldsymbol{V}_s \boldsymbol{h} - \boldsymbol{v}_s\|) \tag{12}$$

where $\boldsymbol{v}_0 \in \mathbb{R}^N$, $\boldsymbol{V}_0 \in \mathbb{R}^{N\times N}$ is a symmetric positive semi-definite matrix, and, for every $s \in \{1, \ldots, S\}$, $\boldsymbol{v}_s \in \mathbb{R}^{P_s}$, $\boldsymbol{V}_s \in \mathbb{R}^{P_s \times N}$, and $\psi_s \colon \mathbb{R} \to \mathbb{R}$. In addition, the following assumptions will be made:

Assumption 1.

(i) $\boldsymbol{R} + \boldsymbol{V}_0$ *is a positive definite matrix.*

(ii) *For every* $s \in \{1, \ldots, S\}$, ψ_s *is a lower-bounded differentiable function and* $\lim_{\substack{t\to 0\\ t\neq 0}} \dot{\psi}_s(t)/t \in \mathbb{R}$, *where* $\dot{\psi}_s$ *denotes the derivative of* ψ_s.

(iii) *For every* $s \in \{1, \ldots, S\}$, $\psi_s(\sqrt{\cdot})$ *is concave on* $[0, +\infty[$.

(iv) *There exists* $\overline{\upsilon} \in [0, +\infty[$ *such that* $(\forall s \in \{1, \ldots, S\})$ $(\forall t \in]0, +\infty[)$ $0 \leqslant \upsilon_s(t) \leqslant \overline{\upsilon}$, *where* $(\forall t \in [0, +\infty[)$ $\upsilon_s(t) = \dot{\psi}_s(t)/t$.[1]

These assumptions are satisfied by a wide class of functions Ψ, in particular quadratic regularization functions, $\ell_2 - \ell_1$ functions, and various forms of smooth $\ell_2 - \ell_0$ functions [6].

Note that, for every $n \in \mathbb{N}^*$, the gradient of F_n is given by

$$(\forall \boldsymbol{h} \in \mathbb{R}^N) \qquad \nabla \mathrm{F}_n(\boldsymbol{h}) = \mathbf{A}_n(\boldsymbol{h})\boldsymbol{h} - \mathbf{c}_n(\boldsymbol{h}) \tag{13}$$

[1] The function is extended by continuity when $t = 0$.

where

$$\boldsymbol{A}_n(\boldsymbol{h}) = \mathbf{R}_n + \boldsymbol{V}_0 + \boldsymbol{V}^\top \mathrm{Diag}\big(\boldsymbol{b}(\boldsymbol{h})\big)\boldsymbol{V} \in \mathbb{R}^{N\times N} \quad (14)$$

$$\mathbf{c}_n(\boldsymbol{h}) = \mathbf{r}_n + \boldsymbol{v}_0 + \boldsymbol{V}^\top \mathrm{Diag}\big(\boldsymbol{b}(\boldsymbol{h})\big)\boldsymbol{v} \in \mathbb{R}^{N} \quad (15)$$

$$\boldsymbol{V} = [\boldsymbol{V}_1^\top \dots \boldsymbol{V}_S^\top]^\top \in \mathbb{R}^{P\times N} \quad (16)$$

$$\boldsymbol{v} = [\boldsymbol{v}_1^\top \dots \boldsymbol{v}_S^\top]^\top \in \mathbb{R}^{P} \quad (17)$$

with $P = P_1 + \cdots + P_S$, and $\boldsymbol{b}(\boldsymbol{h}) = \big(b_i(\boldsymbol{h})\big)_{1\leqslant i\leqslant P} \in \mathbb{R}^P$ is such that $(\forall s \in \{1,\dots,S\})\ (\forall p \in \{1,\dots,P_s\})$

$$b_{P_1+\cdots+P_{s-1}+p}(\boldsymbol{h}) = \upsilon_s(\|\boldsymbol{V}_s\boldsymbol{h} - \boldsymbol{v}_s\|). \quad (18)$$

We have then the following result:

Proposition 1. *Under Assumptions 1(ii)-1(iv), for every $n \in \mathbb{N}^*$ and $\boldsymbol{h} \in \mathbb{R}^N$, a tangent majorant of F_n at $\boldsymbol{h}$ is*

$$(\forall \boldsymbol{h}' \in \mathbb{R}^N) \quad \Theta_n(\boldsymbol{h}', \boldsymbol{h}) = \mathrm{F}_n(\boldsymbol{h}) + \nabla \mathrm{F}_n(\boldsymbol{h})^\top(\boldsymbol{h}' - \boldsymbol{h}) + \frac{1}{2}(\boldsymbol{h}' - \boldsymbol{h})^\top \boldsymbol{A}_n(\boldsymbol{h})(\boldsymbol{h}' - \boldsymbol{h}) \quad (19)$$

where $\boldsymbol{A}_n(\boldsymbol{h})$ is given by (14)*.*

The proposed MM subspace algorithm consists of defining the following sequence of random vectors $(\mathbf{h}_n)_{n\geqslant 1}$:

$$(\forall n \in \mathbb{N}^*) \qquad \mathbf{h}_{n+1} \in \operatorname*{arg\,min}_{\boldsymbol{h}\in \operatorname{span}\mathbf{D}_n} \Theta_n(\boldsymbol{h}, \mathbf{h}_n) \quad (20)$$

where $\operatorname{span}\mathbf{D}_n$ is the vector subspace delineated by the columns of matrix $\mathbf{D}_n \in \mathbb{R}^{N\times M_n}$, and $\mathbf{h}_1$ has to be set to an initial value. For example, we can choose, for every $n \in \mathbb{N}^*$,

$$\mathbf{D}_n = \begin{cases} [-\nabla \mathrm{F}_n(\mathbf{h}_n), \mathbf{h}_n, \mathbf{h}_n - \mathbf{h}_{n-1}] & \text{if } n > 1 \\ [-\nabla \mathrm{F}_n(\mathbf{h}_1), \mathbf{h}_1] & \text{if } n = 1 \end{cases} \quad (21)$$

which yields the 3MG algorithm. Note that a similar choice of subspace can be found in optimization algorithms such as TWIST [21]. A common assumption for subspace algorithms which will be adopted subsequently is that $\nabla \mathrm{F}_n(\mathbf{h}_n)$ belongs to $\operatorname{span}\mathbf{D}_n$.

3.2. Recursive MM strategy

By setting, for every $n \in \mathbb{N}$, $\mathbf{h}_{n+1} = \mathbf{D}_n\mathbf{u}_n$ where $\mathbf{u}_n$ is an $\mathbb{R}^{M_n}$-valued random vector, we deduce from (13), (19) and (20) that

$$\begin{aligned}\mathbf{u}_n &= \mathbf{B}_n^\dagger \mathbf{D}_n^\top \big(\mathbf{A}_n(\mathbf{h}_n)\mathbf{h}_n - \nabla \mathbf{F}_n(\mathbf{h}_n)\big) \\ &= \mathbf{B}_n^\dagger \mathbf{D}_n^\top \mathbf{c}_n(\mathbf{h}_n)\end{aligned} \quad (22)$$

where

$$\mathbf{B}_n = \mathbf{D}_n^\top \mathbf{A}_n(\mathbf{h}_n)\mathbf{D}_n \quad (23)$$

and $(\cdot)^\dagger$ is the pseudo-inverse operation. It is important to note that, as $\mathbf{B}_n$ is of dimension $M_n \times M_n$ where M_n is small (typically $M_n = 3$), this pseudo-inversion is not costly. This constitutes the main advantage of the proposed approach.

Let us now introduce the intermediate variables:

$$(\forall n \in \mathbb{N}^*) \qquad \mathbf{D}_n^{\mathbf{R}} = \mathbf{R}_n\mathbf{D}_n \in \mathbb{R}^{N\times M_n} \quad (24)$$

$$\mathbf{D}_n^{\boldsymbol{V}_0} = \boldsymbol{V}_0\mathbf{D}_n \in \mathbb{R}^{N\times M_n} \quad (25)$$

$$\mathbf{D}_n^{\boldsymbol{V}} = \boldsymbol{V}\mathbf{D}_n \in \mathbb{R}^{P\times M_n} \quad (26)$$

$$\mathbf{D}_n^{\mathbf{A}} = \mathbf{A}_{n+1}(\mathbf{h}_{n+1})\,\mathbf{D}_n \in \mathbb{R}^{N\times M_n}. \quad (27)$$

By using (8), (9), (13) (14), (15), (22), (23), and by performing recursive updates of $(\mathbf{r}_n)_{n\geqslant 1}$ and $(\mathbf{R}_n)_{n\geqslant 1}$, Algorithm 1 is obtained.

Algorithm 1 Stochastic MM subspace method

$\mathbf{r}_0 = \mathbf{0}, \mathbf{R}_0 = \mathbf{0}$
Initialize $\mathbf{D}_0, \mathbf{u}_0$
$\mathbf{h}_1 = \mathbf{D}_0\mathbf{u}_0, \mathbf{D}_0^{\mathbf{R}} = \mathbf{0}, \mathbf{D}_0^{\boldsymbol{V}_0} = \boldsymbol{V}_0\mathbf{D}_0, \mathbf{D}_0^{\boldsymbol{V}} = \boldsymbol{V}\mathbf{D}_0$
For all $n = 1, \dots$
- $\mathbf{r}_n = \mathbf{r}_{n-1} + \frac{1}{n}(\mathbf{X}_n\mathbf{y}_n - \mathbf{r}_{n-1})$
- $\mathbf{c}_n(\mathbf{h}_n) = \mathbf{r}_n + \boldsymbol{v}_0 + \boldsymbol{V}^\top \mathrm{Diag}\big(\boldsymbol{b}(\mathbf{h}_n)\big)\boldsymbol{v}$
- $\mathbf{D}_{n-1}^{\mathbf{A}} = (1 - \frac{1}{n})\mathbf{D}_{n-1}^{\mathbf{R}} + \frac{1}{n}\mathbf{X}_n(\mathbf{X}_n^\top\mathbf{D}_{n-1}) + \mathbf{D}_{n-1}^{\boldsymbol{V}_0} + \boldsymbol{V}^\top \mathrm{Diag}\big(\boldsymbol{b}(\mathbf{h}_n)\big)\mathbf{D}_{n-1}^{\boldsymbol{V}}$
- $\nabla \mathrm{F}_n(\mathbf{h}_n) = \mathbf{D}_{n-1}^{\mathbf{A}}\mathbf{u}_{n-1} - \mathbf{c}_n(\mathbf{h}_n)$
- $\mathbf{R}_n = \mathbf{R}_{n-1} + \frac{1}{n}(\mathbf{X}_n\mathbf{X}_n^\top - \mathbf{R}_{n-1})$
- Set $\mathbf{D}_n$ using $\nabla \mathrm{F}_n(\mathbf{h}_n)$
- $\mathbf{D}_n^{\mathbf{R}} = \mathbf{R}_n\mathbf{D}_n, \mathbf{D}_n^{\boldsymbol{V}_0} = \boldsymbol{V}_0\mathbf{D}_n, \mathbf{D}_n^{\boldsymbol{V}} = \boldsymbol{V}\mathbf{D}_n$
- $\mathbf{B}_n = \mathbf{D}_n^\top\big(\mathbf{D}_n^{\mathbf{R}} + \mathbf{D}_n^{\boldsymbol{V}_0} + \boldsymbol{V}^\top \mathrm{Diag}\big(\boldsymbol{b}(\mathbf{h}_n)\big)\mathbf{D}_n^{\boldsymbol{V}}\big)$
- $\mathbf{u}_n = \mathbf{B}_n^\dagger\mathbf{D}_n^\top\mathbf{c}_n(\mathbf{h}_n)$
- $\mathbf{h}_{n+1} = \mathbf{D}_n\mathbf{u}_n$

3.3. Complexity

Since M_n is small, the complexity of a direct implementation of this algorithm, evaluated in terms of multiplications at iteration n, is of the order of

$$N\big(P(3M_n + 1) + N(4M_n + Q)/2\big)$$

when N is large. However, this complexity can be reduced if matrices $\boldsymbol{V}_0$ or $\boldsymbol{V}$ have a specific structure. In particular, if they are null matrices, the algorithm has the same order of complexity as the classical recursive least squares algorithm. Since the criterion then reduces to a quadratic function, Sherman-Morrison-Woodbury formula can be used in order to calculate iteratively the minimizer on the whole space in an efficient manner. The computational complexity can also be reduced by taking advantage of the specific form of matrices $(\mathbf{D}_n)_{n\geqslant 1}$. For example, if the subspace is chosen according to (21), for every $n > 1$,

$$\mathbf{D}_n^{\boldsymbol{V}} = [-\boldsymbol{V}\nabla \mathrm{F}_n(\mathbf{h}_n), \boldsymbol{V}\mathbf{h}_n, \boldsymbol{V}\mathbf{h}_n - \boldsymbol{V}\mathbf{h}_{n-1}]. \quad (28)$$

On the other hand,

$$\boldsymbol{V}\mathbf{h}_n = \boldsymbol{V}\mathbf{D}_{n-1}\mathbf{u}_{n-1} = \mathbf{D}_{n-1}^{\boldsymbol{V}}\mathbf{u}_{n-1}, \tag{29}$$

which shows that a recursive formula holds to compute the last two components of $\mathbf{D}_n^{\boldsymbol{V}}$ in (28). The initial complexity of $3PN$ multiplications is thus reduced to $(P+3)N$. Similar recursive procedures can be employed to compute $(\mathbf{D}_n^{\boldsymbol{V}_0})_{n>1}$ and $(\mathbf{D}_n^{\mathbf{R}_n})_{n>1}$.

4. APPLICATION TO 2D FILTER IDENTIFICATION

4.1. Problem statement

We now demonstrate the efficiency of the proposed stochastic algorithm in a filter identification problem. Consider the following observation model:

$$\boldsymbol{y} = S(\overline{\boldsymbol{h}})\boldsymbol{x} + \boldsymbol{w}, \tag{30}$$

where $\boldsymbol{x} \in \mathbb{R}^L$ and $\boldsymbol{y} \in \mathbb{R}^L$ represent the original and degraded version of a given image, $\overline{\boldsymbol{h}} \in \mathbb{R}^N$ is the vectorized version of an unknown two-dimensional blur kernel, S is the linear operator which maps the kernel to its associated Hankel-block Hankel matrix form, and $\boldsymbol{w} \in \mathbb{R}^L$ represents a realization of an additive noise. When the images $\boldsymbol{x}$ and $\boldsymbol{y}$ are of very large scale, finding an estimate $\widehat{\boldsymbol{h}} \in \mathbb{R}^N$ of the blur kernel can be very memory consuming, and one can expect good estimation performance by learning the blur kernel through a sweep of blocks of the dataset.

Let us denote by $\boldsymbol{X} \in \mathbb{R}^{L\times N}$ the matrix such that $S(\boldsymbol{h})\boldsymbol{x} = \boldsymbol{X}\boldsymbol{h}$. Then, we propose to define $\widehat{\boldsymbol{h}}$ as a solution to (1), where, for all $n \in \mathbb{N}^*$, $\mathbf{y}_n \in \mathbb{R}^Q$ and $\mathbf{X}_n^\top \in \mathbb{R}^{Q\times N}$, are subparts of $\boldsymbol{y}$ and $\boldsymbol{X}$, respectively, corresponding to $Q \in \{1,\ldots,L\}$ lines of this vector/matrix. For the regularization term Ψ, we consider, for every $s \in \{1,\ldots,N\}$ ($S = N$), an isotropic penalization on the gradient between neighboring coefficients of the blur kernel, i.e., $P_s = 2$ and $\boldsymbol{V}_s = \left[\boldsymbol{\Delta}_s^{\mathrm{h}}\ \boldsymbol{\Delta}_s^{\mathrm{v}}\right]^\top$, where $\boldsymbol{\Delta}_s^{\mathrm{h}} \in \mathbb{R}^N$ (resp. $\boldsymbol{\Delta}_s^{\mathrm{v}} \in \mathbb{R}^N$) is the horizontal (resp. vertical) gradient operator applied at pixel s. The smoothness of $\boldsymbol{h}$ is then enforced by choosing, for every $s \in \{1,\ldots,S\}$ and $u \in \mathbb{R}$, $\psi_s(u) = \lambda\sqrt{1+u^2/\delta^2}$ with $(\lambda,\delta) \in]0,+\infty[^2$. Finally, in order to guarantee the existence of a unique minimizer, the strong convexity of F is imposed by taking $\boldsymbol{v}_0 = \mathbf{0}$ and $\boldsymbol{V}_0 = \tau\boldsymbol{I}_N$, where τ is a small positive value (typically $\tau = 10^{-10}$).

4.2. Simulation results

The original image, presented in Figure 1(a), is the `San Diego` image, of size 1024×1024 pixels, available at http://sipi.usc.edu/database/. The original blur kernel $\overline{\boldsymbol{h}}$ with size 21×21, and the resulting blurred image, which has been corrupted with a zero-mean Gaussan noise with standard deviation $\sigma = 0.03$ (blurred signal-to-noise ratio equal to 24.8 dB), are displayed in Figures 1(b)(c). Figure 1(d) presents the estimated kernel, using the proposed stochastic algorithm with the subspace given by (21). Parameters (λ,δ) were adjusted so as to minimize the normalized root mean square estimation error, here equal to 0.087. Figure 2 illustrates the variations of the estimation error with respect to the computation time for the proposed algorithm, the SGD algorithm with a decreasing stepsize proportional to $n^{-1/2}$, and the regularized dual averaging (RDA) method with a constant stepsize from [15], when running tests on an Intel(R) Core(TM) i7-3520M @ 2.9GHz using a Matlab 7 implementation. Note that for the latter two algorithms, the stepsize parameter was optimized manually so as to obtain the best performance in terms of convergence speed. Finally, note that stochastic 3MG and RDA algorithms were observed to provide asymptotically the same estimation quality, whatever the size of the blocks. In this example, the best trade-off in terms of convergence speed is obtained for $Q = 64 \times 64$.

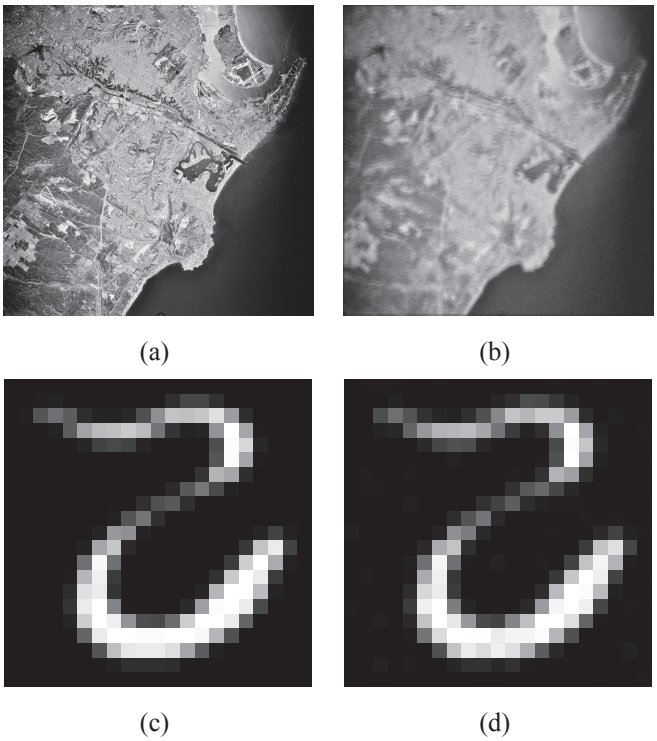

Fig. 1. (a) Original image. (b) Blurred and noisy image. (c) Original blur kernel. (d) Estimated blur kernel, with relative error 0.087.

5. CONCLUSION

In this work, we have proposed a stochastic MM Memory Gradient algorithm for on-line penalized least squares estimation problems. The method makes it possible to use large-size datasets the second-order moments of which are not known a

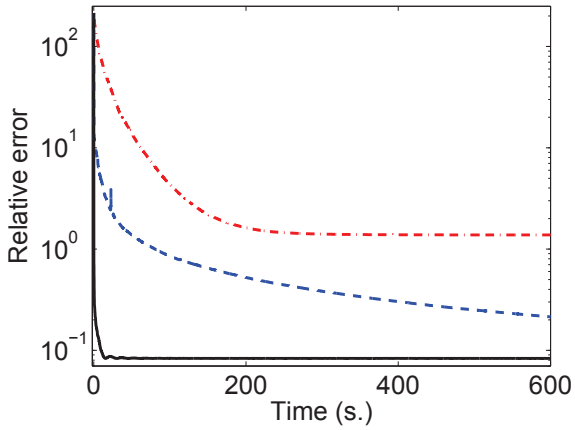

Fig. 2. Comparison of stochastic 3MG algorithm (solid black line), SGD algorithm with decreasing stepsize $\propto n^{1/2}$ (dashed-dotted red line) and RDA algorithm with constant stepsize (dashed blue line).

priori. We have shown that the proposed algorithm is of the same order of complexity as the classical RLS algorithm and that its computational cost can be even reduced by taking advantage of specific forms of the search subspace. The good numerical performance of the proposed algorithm has been demonstrated in the context of 2D filter identification for large size images. In our future work, a theoretical analysis of the convergence properties of the proposed method will be conducted. In addition, we plan to apply this technique to system identification or inverse modeling using adaptive filters.

6. REFERENCES

[1] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.

[2] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. H. Bauschke, R. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, Eds., pp. 185–212. Springer-Verlag, New York, 2010.

[3] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "An augmented Lagrangian aproach to the constrained optimization formulation of imaging inverse problems," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 681–695, 2011.

[4] M. Zibulevsky and M. Elad, "$\ell_2 - \ell_1$ optimization in signal and image processing," *IEEE Signal Process. Mag.*, vol. 27, pp. 76–88, May 2010.

[5] E. Chouzenoux, J. Idier, and S. Moussaoui, "A majorize-minimize subspace strategy for subspace optimization applied to image restoration," *IEEE Trans. Image Process.*, vol. 20, no. 18, pp. 1517–1528, Jun. 2011.

[6] E. Chouzenoux, A. Jezierska, J.-C. Pesquet, and H. Talbot, "A majorize-minimize subspace approach for ℓ_2-ℓ_0 image regularization," *SIAM J. Imag. Sci.*, vol. 6, no. 1, pp. 563–591, 2013.

[7] A. Florescu, E. Chouzenoux, J.-C. Pesquet, P. Ciuciu, and S. Ciochina, "A majorize-minimize memory gradient method for complex-valued inverse problem," *Signal Process.*, vol. 103, pp. 285–295, Oct. 2014, Special issue on Image Restoration and Enhancement: Recent Advances and Applications.

[8] S. O. Haykin, *Adaptive Filter Theory*, Prentice Hall, New Jersey, USA, 4th edition, 2002.

[9] L. Bottou, "Stochastic learning," in *Advanced Lectures on Machine Learning*, O. Bousquet and U. von Luxburg, Eds., Lecture Notes in Artificial Intelligence, LNAI 3176, pp. 146–168. Springer Verlag, Berlin, 2004.

[10] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM J. Control Optim.*, vol. 30, no. 4, pp. 838–855, 1992.

[11] F. Bach and E. Moulines, "Non-asymptotic analysis of stochastic approximation algorithms for machine learning," in *Proc. Ann. Conf. Neur. Inform. Proc. Syst.*, Granada, Spain, Dec. 12 - 17 2011, pp. x–x+8.

[12] C. Paleologu, J. Benesty, and S. Ciochină, *Sparse adaptive filters for echo cancellation*, Synthesis Lectures on Speech and Audio Processing. Morgan and Claypool, San Rafael, USA, 2010.

[13] Y. Murakami, M. Yamagishi, M. Yukawa, and I. Yamada, "A sparse adaptive filtering using time-varying soft-thresholding techniques," in *Proc. Int. Conf. Acoust., Speech Signal Process.*, Dallas, Texas, Mar. 14-19 2010, pp. 3734–3737.

[14] D. Angelosante, J. A. Bazerque, and G. B Giannakis, "Online adaptive estimation of sparse signals: where RLS meets the ℓ_1 -norm," *IEEE Trans. Signal Process.*, pp. 3436–3447, Jul. 2010.

[15] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," *J. Mach. Learn. Res.*, vol. 11, pp. 2543–2596, Oct. 2010.

[16] J. Mairal, "Stochastic Majorization-Minimization algorithms for large-scale optimization," in *Proc. Adv. Conf. Neur. Inform. Proc. Syst.*, Lake Tahoe, Nevada, Dec. 5-8 2013, pp. x–x+8.

[17] J. R. Birge, X. Chen, L. Qi, and Z. Wei, "A stochastic Newton method for stochastic quadratic programs with recourse," 1995, technical report, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.49.-4279.

[18] A. Bordes, L. Bottou, and P. Gallinari, "SGD-QN: Careful quasi-Newton stochastic gradient descent," *J. Mach. Learn. Res.*, vol. 10, pp. 1737–1754, Jul. 2009.

[19] J. Yu, S. V. N. Vishwanathan, S. Günter, and N. N. Schraudolph, "A stochastic quasi-Newton method for online convex optimization," *J. Mach. Learn. Res.*, vol. 11, pp. 1145–1200, Mar. 2010.

[20] R. H. Byrd, S. L. Hansen, J. Nocedal, and Y. Singer, "A stochastic quasi-Newton method for large-scale optimization," 2014, technical report, http://arxiv.org/abs/1401.7020.

[21] J. M. Bioucas-Dias and M. A. T. Figueiredo, "A new TwIST: two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2992–3004, Dec. 2007.