REGULARIZED COMPRESSION OF A NOISY BLURRED IMAGE

Paola Favati[1], Grazia Lotti[2], Ornella Menchi[3] and Francesco Romani[4]

[1]IIT-CNR, Pisa, Italy
[2,3,4]Dipartimento di Matematica, Università di Parma, Parma, Italy

ABSTRACT

Both regularization and compression are important issues in image processing and have been widely approached in the literature. The usual procedure to obtain the compression of an image given through a noisy blur requires two steps: first a deblurring step of the image and then a factorization step of the regularized image to get an approximation in terms of low rank nonnegative factors. We examine here the possibility of swapping the two steps by deblurring directly the noisy factors or partially denoised factors. The experimentation shows that in this way images with comparable regularized compression can be obtained with a lower computational cost.

KEYWORDS

Image Regularization, Image Compression, Nonnegative Matrix Factorization

1. INTRODUCTION

Let X^* be an $n\times n$ matrix whose entries represent an image and let B^* represent the corresponding blurred image generated by an imaging system. Denoting by $\boldsymbol{x}^*=\mathrm{vec}(X^*)$ and $\boldsymbol{b}^*=\mathrm{vec}(B^*)$ the N-vectors (with $N=n^2$) which store columnwise the pixels of X^* and B^*, the imaging system is represented by an $N\times N$ matrix A such that

$$A\boldsymbol{x}^* = \boldsymbol{b}^*. \qquad (1)$$

In general, the vector $\boldsymbol{b}^*$ is not exactly known because it is affected by noise due to the fluctuations in the counting process of the acquisition of the image, which obeys to Poisson statistics, and to the readout imperfections of the recording device, which are Gaussian distributed. Hence we assume that the available recorded image is B with $\mathrm{vec}(B)=\boldsymbol{b}=\boldsymbol{b}^*+\boldsymbol{\eta}$. The noise $\boldsymbol{\eta}$ is obviously unknown, even if in general its level $\eta=\|\boldsymbol{\eta}\|_2/\|\boldsymbol{b}\|_2$ can be roughly estimated. Model (1) so becomes

$$A\boldsymbol{x} = \boldsymbol{b}. \qquad (2)$$

Reconstructing $\boldsymbol{x}^*$ given $\boldsymbol{b}$ is an ill-posed problem, because A may be severely ill-conditioned. This kind of problem arises for example in the reconstruction of astronomical images taken by a telescope and of medical and microscopy images, where the entry $a_{i,j}$ of A measures the fraction of the light intensity of the ith pixel of $\boldsymbol{x}^*$ which arrives at the jth pixel of $\boldsymbol{b}^*$. The quantities involved in the problem, i.e. A, $\boldsymbol{x}^*$, $\boldsymbol{b}^*$ and $\boldsymbol{b}$, are assumed componentwise nonnegative and it is reasonable to expect a nonnegative approximation of $\boldsymbol{x}^*$ when solving (2). This is obtained by imposing nonnegativity constraints, looking for a *constrained least squares* approximation of $\boldsymbol{x}^*$ given by

DOI:10.5121/ijcsa.2016.6601

$$\boldsymbol{x}_{ls} = \arg\min \phi(\boldsymbol{x}),\ \text{under}\ \boldsymbol{x} \geq 0,\ \text{where}\ \phi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{b} - A\boldsymbol{x}\|_2^2$$

Because of the ill-conditioning of A and of the presence of the noise, when system (2) has a large dimension the solution $\boldsymbol{x}_{ls}$ may be quite different from $\boldsymbol{x}^*$, and a regularization method must be employed.

Iterative methods, coupled with suitable strategies for enforcing non negativity, may be used as regularization methods if they enjoy the semi convergence property. According to this property, the vectors computed in the first iterations are minimally affected by the noise and move toward the solution $\boldsymbol{x}^*$ but, going on with the iterations, the computed vectors are progressively contaminated by the noise and move away from $\boldsymbol{x}^*$ toward $\boldsymbol{x}_{ls}$. A vector which is close to $\boldsymbol{x}^*$ is assumed as the regularized solution $\boldsymbol{x}_{reg}$ of (2). A good terminating procedure is hence needed to detect the correct index where to stop the iteration.

Once a regularized solution $\boldsymbol{x}_{reg}$ has been found, one can apply a compression technique to the $n \times n$ matrix X_{reg} obtained by partitioning column wise $\boldsymbol{x}_{reg}$, in order to reduce the storage required for saving it. This second step can be accomplished by using a dimensional reduction technique, for example by computing a low rank nonnegative approximation of X_{reg}. Since nonnegativity is an essential issue in our problem, we refer here to the Nonnegative Matrix Factorization (**NMF**) which computes an approximation of a nonnegative matrix in the form of the product of two low rank nonnegative factors W and H, i.e. $X_{reg} \approx W\,H$.

Besides this two steps procedure, other approaches can be followed to get low rank factorized approximations of X^*. A simple one that we consider here proceeds directly with the data A and $\boldsymbol{b}$ without first computing X_{reg} and combines regularization and factorization steps. Another one, that we also take into consideration following [1], factorizes B with penalty constraints in order to obtain partially denoised factors, which are successively refined involving the action of the blurring matrix.

In this paper we formalize these three procedures with the aim of testing and comparing their ability to produce acceptable regularized compressed approximations of X^*. The outline of the paper is the following: some preliminaries on **NMF** applied to a general matrix are given in Section 2. Then in Section 3 the procedures are described, with details on the use of **FFT** to perform the matrix by vector products and to implement the generalized cross validation (**GCV**) for the stopping rules when the blurring matrix has circulant structure. Finally, Section 4 presents and analyzes the results of the numerical experiments.

2.NOTATION

For a given $n_1 \times n_2$ matrix M, we denote by $\boldsymbol{m}$=vec(M) the vector of $n_1\, n_2$ components which stores columnwise the elements of M, by $\boldsymbol{m}^{(j)}$, j=1, …, n_2 the jth column of M and by

$$m_{i,j} = (\mathrm{M})_{i,j} = \boldsymbol{m}^{(j)}{}_i = (m)_r$$

the (i,j)th entry of M, with $r=(j-1)n_1+i$. Moreover we denote $\boldsymbol{m}'$=vec(M^{T}). In the following the pointwise multiplication and division between vectors will be denoted with ×. and /. respectively. With $\boldsymbol{e}_i$ we denote the ith canonical vector of suitable length.

2.1 THE NONNEGATIVE MATRIX FACTORIZATION (NMF)

Before dealing with our problem, it is useful to furnish some generalities on the problem of the nonnegative matrix factorization.

The best factorization of a matrix X in terms of the Frobenius norm is achieved by the Singular Value Decomposition (**SVD**) $X=U\Sigma V^{\mathrm{T}}$. The best k-rank approximation of X is computed by suitably truncating the **SVD**, but the factors so obtained have in general negative components even if X is nonnegative. Hence a different approach must be followed when non negativity is required.

To fulfil the non negativity requirement, a Nonnegative Matrix Factorization (**NMF**) can be applied. Originally **NMF** was first proposed in [2] as a dimensional reduction technique for data analysis and from then on it has received much attention.

Let R_+^n be the n-dimensional space of vectors with nonnegative components. Given a matrix X of n columns $\boldsymbol{x}^{(i)} \in \mathrm{R}_+^n$, for i=1, …, n and an integer $k << n$, **NMF** looks for two low-rank matrices $W \in \mathrm{R}_+^{n\times k}$ and $H \in \mathrm{R}_+^{k\times n}$ which solve the problem

$$\min \psi(W,H), \text{ under } W,H \geq 0, \quad \text{with } \psi(W,H) = \frac{1}{2}\|X - WH\|_F^2 \tag{3}$$

where F denotes the Frobenius norm. In this way, each column $\boldsymbol{x}^{(i)}$ is approximated by the linear combination with nonnegative coefficients $h_{r,i}$ of the vectors $\boldsymbol{w}^{(r)}$ for r=1, …, k. Problem (3) is non convex and finding its global minimum is NP-hard, so one only looks for local minima. Most non convex optimization algorithms, like for example standard gradient methods, guarantee the stationarity of the limit points but suffer from slow convergence. Newton-like algorithms have a better rate of convergence but are more expensive.

The following alternating nonnegative least squares (**ANLS**) approach, which belongs to the block coordinate descent framework [3] of nonlinear optimization, has shown to be very efficient. Its scheme is very simple: one of the factors, say W, is initialized to an initial matrix W_0 with nonnegative entries and the matrix $H \in \mathrm{R}_+^{k\times n}$ which realizes the $\min\|X - W_0H\|_F^2$ is computed. The matrix so obtained is assumed as H_1 and a new matrix $W_1 \in \mathrm{R}_+^{n\times k}$ which attains the $\min\|X - WH_1\|_F^2$ is computed, and so on, updating W and H alternatively.
In practice the algorithm computes

$$\begin{cases} H_{\nu+1} = \arg\min \psi(W_\nu, H)\ , \text{ under } H \geq 0, \quad \text{given } W_\nu \\ W_{\nu+1} = \arg\min \psi(W, H_{\nu+1}), \text{ under } W \geq 0, \text{given } H_{\nu+1} \end{cases} \tag{4}$$

for ν=0,1, …, until a suitable condition verifies that a local minimum of the object function ψ(W,H) of (3) has been sufficiently well approximated. The iteration stops when

$$\frac{|\varepsilon_\nu - \varepsilon_{\nu+1}|}{\varepsilon_\nu} < tol \text{ where } \varepsilon_\nu = \frac{\|X - W_\nu H_\nu\|_F}{\|X\|_F} \tag{5}$$

for a pre assigned tolerance tol.

The choice of the initial matrix W_0 may be critical, due to the fact that only a local minimum is expected, which obviously depends on this choice. Typically, the algorithm is run several times with different initial random matrices and the run with the lowest error ε_ν is chosen. Both problems (4) can be cast in the following form

$$\min \varphi(Z), \text{ under } Z \geq 0, \text{ where } \varphi(Z) = \frac{1}{2}\|U - VZ\|_F^2 \qquad (6)$$

where $U \in \mathrm{R}_+^{n\times n}$ and $V \in \mathrm{R}_+^{n\times k}$ are given and $Z \in \mathrm{R}_+^{k\times n}$ has to be computed. Specifically, U=X, V=W_ν, Z=H for the first problem of (4) and U=X^T, V=$H^T{}_{\nu+1}$, Z=W^T for the second problem of (4). The gradient and the Hessian of φ are

$$G(Z) = V^T VZ - V^T U \text{ and } Q = V^T V.$$

Although the original problem (3) is non convex, problem (6) is convex and can be easily dealt with by applying any procedure for constrained quadratic optimization, for example an Active-Set-like method [4, 5]. According to [6], the requirement that the non negativity constraints are exactly satisfied at each step is necessary for convergence but makes the algorithm rather slow at large dimensions. Moreover, in our context where the presence of the noise must be accounted for, computing the exact solution of problems (4) is not worthwhile.

Faster algorithms have been suggested to compute iteratively approximate solutions with inexact methods like gradient descent, Newton-type methods modified to satisfy the non negativity constraints and coordinate descent methods. In the experimentation we have used the Greedy Coordinate Descent method (**GCD**) specifically designed in [7] for solving problems of the form (6) which proceeds as follows. Starting with an initial matrix Z_0 (a possible choice is $Z_0 = O$, which ensures that $G(Z_0) \leq O$), the matrices Z_μ μ=0,1, …, are computed iteratively with

$$Z_{\mu+1} = Z_\mu + \sum_{j=1}^{n} \sum_{i\in I(j)} s_{i,j}\, e_i e_j^T$$

where the indices i in the set $I(i)$ and their order are not defined a-priori, but are determined by running conditions. The coefficients $s_{i,j}$ are determined by minimizing function φ on the dyad $e_i e_j^T$. For any j the indices i are chosen according to a maximal decrease criterium applied to φ. The code for **GCD** can be found as Algorithm 1 in [7].

The dominant part of the computational cost of solving each problem (6) using **GCD** is given by the cost of updating the gradient, in particular by the cost $\gamma_{kn} = k\, n^2$ of computing the product VZ. Hence the cost of computing the **NMF** factorization of the matrix X is $\gamma_{fac} = 2\ \gamma_{kn}\ it_{fac}$, where it_{fac} is the number of iterations required to satisfy (5).

3. The Problem

The problem we consider is the following: find a compressed approximation of the original n×n image X* by solving in a regularizing way the system

$$A\boldsymbol{x} = \boldsymbol{b}. \qquad (7)$$

By the term "compression" we mean an approximation given by the product of two k-rank matrices, with k an integer such that $k << n$. Write the N×N given matrix A, with N=n^2, in block form

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} & \cdots & A_{1,n} \\ A_{2,1} & A_{2,2} & \ldots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \cdots & A_{n,n} \end{bmatrix}, \text{ with } A_{i,j} \in \mathbb{R}_+^{n\times n}$$

System (7) becomes

$$\sum_{j=1}^{n} A_{i,j}\boldsymbol{x}^{(j)} = \boldsymbol{b}^{(i)}, \quad i = 1,\ldots,n \tag{8}$$

Due to the generally large size of the problem, some structure must be assumed for *A*. When the problem is modeled by a Fredholm integral equation defined by a point spread function (**PSF**) invariant with respect to translations on a bounded support, *A* is a 2-level Toeplitz matrix, i.e.

$$A_{i,j} = A_{i-1,j-1}, \left(A_{i,j}\right)_{r,s} = \left(A_{i,j}\right)_{r-1,s-1},$$

Moreover, if periodic boundary conditions can be safely imposed, *A* becomes a 2-level circulant matrix. Hence $A_{i,1} = A_{i-1,n}$for any *i* and the same structure holds internally for each block.

We assume as regularized solution $\boldsymbol{x}_{reg}$ an approximation of the least squares solution $\boldsymbol{x}_{ls}$ obtained by minimizing the function

$$\phi(\boldsymbol{x}) = \frac{1}{2}\left\|\boldsymbol{b} - A\boldsymbol{x}\right\|_2^2 \tag{9}$$

under nonnegativity constraints. The gradient and the Hessian of $\phi(\boldsymbol{x})$ are

$$\text{grad}_x\phi(\boldsymbol{x}) = A^TA\boldsymbol{x} - A^T\boldsymbol{b}, \qquad \text{Hess}\left(\phi(\boldsymbol{x})\right) = A^TA \tag{10}$$

Since the Hessian is positive semidefinite, $\phi(\boldsymbol{x})$ is convex. Its minimum points solve the system $\text{grad}_x\phi(\boldsymbol{x}) = 0$, i.e. the so-called *normal equations* $A^TA\boldsymbol{x} = A^T\boldsymbol{b}$. The minimizer of $\phi(\boldsymbol{x})$ under nonnegativity constraints can be approximated by any method for constrained quadratic optimization. We suggest to use an iterative semiconvergent method to ensure the regularization of the solution with starting point $A^T\boldsymbol{b}$. The number of performed iterations is denoted by it_{sol}.

3.1. The Factorizing Systems

Let *X* be the *n*×*n* matrix obtained by partitioning columnwise $\boldsymbol{x}$ and set *X*=*WH*, with $W \in \mathbb{R}_+^{n\times k}$ and $H \in \mathbb{R}_+^{k\times n}$. Replacing the relation $\boldsymbol{x}^{(j)} = W\boldsymbol{h}^{(j)}, \quad j = 1,\ldots,n$, in (8) we get the system

$$\sum_{j=1}^{n} A_{i,j}W\boldsymbol{h}^{(j)} = \boldsymbol{b}^{(i)}, \quad i = 1,\ldots,n \tag{11}$$

which in components is written as

$$\sum_{j=1}^{n}\sum_{t=1}^{k}\left[\sum_{s=1}^{n}\left(A_{i,j}\right)_{r,s} w_{s,t}\right] h_{t,j} = b_{r,i}, \quad r,i = 1,\ldots,n \tag{12}$$

and equivalently

$$A(I_n \otimes W)\boldsymbol{h} = \boldsymbol{b} \tag{13}$$

System (13) allows computing the columns of H if W is assigned.

We now perform a permutation of A using the N×N matrix Π with blocks of the form $\Pi_{i,j} = e_j e_i^T$. The element $\left(A'_{r,s}\right)_{i,j}$ of the matrix A'=ΠAΠ^T is equal to $\left(A_{i,j}\right)_{r,s}$. System (12) becomes

$$\sum_{s=1}^{n}\sum_{t=1}^{k}\left[\sum_{j=1}^{n}\left(A'_{r,s}\right)_{i,j} h_{t,j}\right] w_{s,t} = b_{r,i}, \quad i,r = 1,\mathrm{K}\ ,n \tag{14}$$

and equivalently

$$A'(I_n \otimes H^T)\boldsymbol{w}' = \boldsymbol{b}' \tag{15}$$

System (15) allows computing the rows of W if H is assigned. The constrained least squares solutions of the two systems (13) and (15) are

$$\boldsymbol{h}_{ls} = \arg\min \phi_h(W,\boldsymbol{h}), \text{ under } \boldsymbol{h} \geq 0, \text{ where } \phi_h(W,\boldsymbol{h}) = \frac{1}{2}\left\|\boldsymbol{b} - A(I_n \otimes W)\boldsymbol{h}\right\|_2^2 \tag{16}$$

and

$$\boldsymbol{w}_{ls} = \arg\min \phi_w(H,\boldsymbol{w}), \text{ under } \boldsymbol{w} \geq 0, \text{ where } \phi_w(H,\boldsymbol{w}) = \frac{1}{2}\left\|\boldsymbol{b}' - A'(I_n \otimes H^T)\boldsymbol{w}'\right\|_2^2 \tag{17}$$

Problems (16) and (17) can be inserted into an **ANLS** scheme. In the first step of this alternating inner-outer method we assume that W has been fixed and solve with respect to H, in the second step H is fixed and W is obtained. Thus, starting with initial (W_0, H_0), a sequence (W_ν, H_ν), ν=0,1, … is computed iteratively, with $\boldsymbol{w}_\nu$ = vec(W_ν), $\boldsymbol{h}_\nu$ = vec(H_ν) and

$$\begin{cases} \boldsymbol{h}_{\nu+1} = \arg\min \phi_h(W_\nu,\boldsymbol{h}), \text{ under } \boldsymbol{h} \geq 0, \\ \boldsymbol{w}_{\nu+1} = \arg\min \phi_w(H_{\nu+1},\boldsymbol{w}), \text{ under } \boldsymbol{w} \geq 0, \end{cases} \tag{18}$$

Both problems (18) can be cast in the form

$$\min \varphi(\boldsymbol{z}), \quad \text{under } \boldsymbol{z} \geq 0, \quad \text{with } \varphi(\boldsymbol{z}) = \frac{1}{2}\left\|M\boldsymbol{z} - \boldsymbol{u}\right\|_2^2 \tag{19}$$

where $M = A(I_n \otimes W_\nu)$, $\boldsymbol{z} = \boldsymbol{h}$, $\boldsymbol{u} = \boldsymbol{b}$ for the first problem of (18) and $M = A'(I_n \otimes H_{\nu+1}^T)$, $\boldsymbol{z} = \boldsymbol{w}'$, $\boldsymbol{u} = \boldsymbol{b}'$ for the second problem of (18). Problem (19) can be non negatively solved by any procedure for constrained quadratic optimization. As starting point we choose (W_0, H_0) a rough nonnegative factorization of the n×n matrix obtained by partitioning column wise $A^T\mathbf{b}$ .Two stopping conditions should be set: one for the method used for solving the inner problem (19) and one for the whole **ANLS** scheme. The total number of inner iterations performed when solving the **ANLS** scheme is denoted by it_{sol}.

3.2. THE PROCEDURES

We formalize now three procedures that we take into consideration for obtaining a compressed approximation of the original image X^* in the form of a k-rank factorization. • *Procedure* **P1**: solve (8) with a method which enjoys the semi convergence property, starting with initial approximation $A^T\boldsymbol{b}$. Partition the solution $\boldsymbol{x}_{reg}$ by column into the n×n matrix X_{reg} and apply

NMF to obtain the factorization $W^{(1)}H^{(1)} \approx X_{reg}$ using (5) with tolerance *tol* as a stopping condition (in the experiments we set $tol = 10^{-3}$ since lower factorization errors in the compressed images are not remarkable at a visual inspection). For solving (8) we use (for comparison purpose) two semiconvergent methods which enjoy nonnegativity feature: Expectation-Maximization (**EM**) [8] and a scaled gradient projection method especially designed for nonnegative regularization (**SGP**) [9].

• *Procedure* **P2**: apply the **ANLS** scheme to (18). Two initial matrices W_0 and H_0 are computed through an incomplete **NMF** of the $n\times n$ matrix obtained by partitioning columnwise $A^T\boldsymbol{b}$, i.e. an **NMF** run for few iterations (in the experiments for only 5 iterations). Both problems (18) are solved using the semiconvergent methods already used for procedure P1. The matrices computed at the end of procedure **P2** are denoted $W^{(2)}$ and $H^{(2)}$.

• *Procedure* **P3**: include a penalty term in the function $\psi(W, H)$ of (3) with the aim of balancing the trade-off between the approximation error and the influence of the noise by enhancing the smoothness of the factors. Since in general the noise acts in producing solutions of (8) of very large magnitude, the function to minimize becomes

$$\psi(W,H)=\frac{1}{2}\left[\|B-WH\|_F^2+\lambda\left(\|W\|_F^2+\|H\|_F^2\right)\right]$$

where λ is a scalar regularization parameter. Then apply NMF factorization to B. The mean of the elements of B is considered a good choice for λ. The penalized factors W_p and H_p so computed must be further treated to cope with the blurring. So the vector $\boldsymbol{w}'_0 = vec(W_p^T)$ is taken as starting point for solving problem (17) with matrix H_p i.e.

$$\min\frac{1}{2}\left\|\boldsymbol{b}'-A'(I_n\otimes H_p^T)\boldsymbol{w}'\right\|_2^2,\ \text{under}\ \boldsymbol{w}\geq 0 \tag{20}$$

The matrix computed by solving (20), denoted $W^{(3)}$, is assumed as first final factor, the second factor being $H^{(3)} = H_p$. The semi convergent methods already used for procedure **P1** are used also here.

The stopping condition used for these procedures is the generalized cross validation (**GCV**) (see Section 3.5) with a tolerance θ which can be tuned according to the aims of the various procedures.

3.3. COMPUTATIONAL COST

Apart the cost of computing the **NMF** factorizations which has been examined at the end of Section 2, the main part of the computational cost of the procedures is due to matrix by vector products. Denoting by γ_{mxv} the cost of this product, a procedure which runs for it_{sol} iterations has a computational cost of order $it_{sol}\ \gamma_{mxv}$. Both the methods we use (**EM** and **SGP**) require two matrix by vector products per iteration, but to this cost we must add the cost γ_{sto} for the test of the stopping condition. Denoting by γ_A the cost of computing $A\boldsymbol{x}$, where A is an $n^2\times n^2$ matrix, for system (8) it is $\gamma_{mxv} = \gamma_A$, while for system (13) (and equivalently for system (15)) γ_{mxv} is the cost of computing $A(I_n \otimes W)\boldsymbol{h}$, which is typically performed by A vec(WH). Then in this case $\gamma_{mxv} = \gamma_A + \gamma_{kn}$ where γ_{kn} is defined at the end of Section 2.

Procedure **P1** consists of a reconstruction phase where system (8) is solved, followed by a factorization phase. The corresponding iteration numbers are it_{sol} and it_{fac}. Also procedure **P2**

consists of two phases, but in this case the factorization phase precedes the direct solution of the systems of the **ANLS** scheme (18). The corresponding iteration numbers are it_{fac} = 5 and it_{sol}. Analogously, for procedure **P3** the two phases are a factorization and the direct solution of (20). The corresponding iteration numbers are it_{fac} and it_{sol}.
The total cost results to be of order

$$\begin{aligned} & it_{sol}\,(2\gamma_A+\gamma_{sto}) + 2\; it_{fac}\,\gamma_{kn} && \text{for } \mathbf{P1} \\ & it_{sol}\,(2\gamma_A+\gamma_{sto}) + (10 + 2\; it_{sol})\,\gamma_{kn} && \text{for } \mathbf{P2} \\ & it_{sol}\,(2\gamma_A+\gamma_{sto}) + 2(it_{fac}+ it_{sol})\,\gamma_{kn} && \text{for } \mathbf{P3} \end{aligned} \qquad (21)$$

3.4. CIRCULANT MATRICES

The procedures described in Section 3.2 can be applied to matrices A of any form. In the case of non sparse unstructured matrices the cost γ_A of the matrix by vector product can be as high as n^4. We consider here the case of circulant matrices for which the product of A and A^T by a vector can be performed using a low cost algorithm based on the fast Fourier transform (**FFT**).

A circulant matrix A of size $N = n^2$ is diagonalized by the Fourier matrix F, whose elements are

$$f_{r,s} = \frac{1}{n}\omega^{rs}, \quad r,s = 0,\ldots n^2-1, \quad \text{with } \omega = \exp(2\pi\boldsymbol{i}/n^2)$$

Denoting by $\boldsymbol{a}^T$ the first row of A, it holds $A = F\ \mathrm{diag}(F\boldsymbol{a})\ F^*$, where F^* is the inverse (i.e. transpose conjugate) of F. Hence the product **z**=$A\boldsymbol{v}$, where $\boldsymbol{v}$ and $\boldsymbol{z}$ are n^2 vector, can be so computed

$$\boldsymbol{a}" = F\boldsymbol{a}, \quad \boldsymbol{v}" = F^*\boldsymbol{v}, \quad \boldsymbol{z} = F(\boldsymbol{a}" \times . \boldsymbol{v}") \qquad (22)$$

To obtain the product $\boldsymbol{z}=A^T\boldsymbol{v}$ it is sufficient to take the first column of A as $\boldsymbol{a}$ or to replace $\boldsymbol{a}''$ by its conjugate $\boldsymbol{a}''^*$ and $\boldsymbol{z}=A^TA$ **v** is computed by $\boldsymbol{z} = F(\boldsymbol{a}''^* \times .\ \boldsymbol{a}''\times .\ \boldsymbol{v}'')$. The multiplications by F or F^* can be efficiently computed by calling an **FFT** routine, which has a computational cost γ_{ft} of order $n^2 \log n$. Then the cost of performing a matrix by vector product when A has a circulant structure is $\gamma_{mxv} = 2\ \gamma_{ft}$.

3.5. GENERALIZED CROSS VALIDATION (GCV)

The execution flow of iterative algorithms is ruled by stopping conditions. In our case, it is not worthwhile to solve problems (18) too accurately. We suggest to use the Generalized Cross Validation method (**GCV**) described in [10, 11]. The performance of **GCV** has been tested in [12, 13], and it resulted that **GCV** is more effective than other stopping rules when information about the entity of the noise is not available. Given a sequence $\boldsymbol{x}_\nu$ generated by an iterative method to approximate the solution of the system $A\boldsymbol{x} = \boldsymbol{b}$, the **GCV** functional is defined by

$$V_\nu = \frac{N\left\|\boldsymbol{b} - A\boldsymbol{x}_\nu\right\|_2^2}{\left(N - \mathrm{trace}(A_\nu)\right)^2} \qquad (23)$$

where the influence matrix A_ν is such that $A_\nu\boldsymbol{b} = A\boldsymbol{x}_\nu$. The minimizer of V_ν gives a good indication of the index where the iteration should be stopped, but in some cases this indication can be misleading because of the almost flat behavior of V_ν near the minimum, so it is suggested to stop the iteration when

$$(V_\nu - V_{\nu+1})/V_\nu < \theta$$

where θ is a pre assigned tolerance. The larger the tolerance θ, the more anticipated the stop.

To apply (23) an estimate of the trace of A_ν must be provided. When the method is applied to a matrix A which has a 2-level circulant structure, it is reasonable to look for an estimate computed through a 2-level circulant approximation T_ν of A_ν such that $A\boldsymbol{x}_\nu \approx T_\nu \boldsymbol{b}$. Denoting by $\boldsymbol{t}_\nu^{\mathrm{T}}$ the first row of T_ν, we have $T_\nu \boldsymbol{b} = F\,\mathrm{diag}(F\,\boldsymbol{t}_\nu)\,F^*\boldsymbol{b}$, hence

$$\boldsymbol{t}_\nu \approx F^*\big(F^*(A\boldsymbol{x}_\nu)./F^*\boldsymbol{b}\big) \quad \text{and} \quad \mathrm{trace}(A_\nu) \approx \sum_i (\boldsymbol{t}_\nu)_{\mathrm{i}} \tag{24}$$

Different ways to estimate the trace of A_ν can be found in [14]. The estimate given in (24) is very simple, but lays open the question of which method can exploit it. A preliminary ad-hoc experimentation on the problems considered in the next section, whose exact solution $\boldsymbol{x}^*$ is known, showed that the index found using (24) was acceptable for both **EM** and **SGP** as long as matrix A was circulant. The indices so obtained were only slightly smaller than the ones corresponding to the minimum error and led to an anticipated stop. Since this outcome is considered favorable in a regularizing context where a too long computation may produce bad solutions, (24) has been used, together with an upper limitation on the number of allowed iterations.

Using (24), the cost of applying **GCV** is $\gamma_{sto} = 2\ \gamma_{ft}$. Replacing γ_{sto} and $\gamma_A = 2\ \gamma_{ft}$ in (21) we have that the computational cost of the three procedures is

$$c_1\ \gamma_{ft} + c_2\ \gamma_{kn} \tag{25}$$

with $c_1 = 6\ it_{sol}$, and $c_2 = 2\ it_{fac}$ for **P1**, $c_2 = (10+2\ it_{sol})$ for **P2**, $c_2 = 2(it_{fac} + it_{sol})$ for **P3**.

4. Numerical Experiments

The numerical experimentation has been conducted with double precision arithmetic. We consider two reference objects X^*, the satellite image [15] and a Hoffman phantom [16], widely used in the literature for testing image deconvolution algorithms. For both images n=256 and k=10. The compressed 10-rank approximations W^*H^* of X^* are generated by means of **NMF**. The compression errors of the images have been preliminarily measured by

$$\varepsilon^* = \frac{\|X^* - W^* H^*\|_F}{\|X^*\|_F}$$

and are ε*=33% for the satellite and ε*=40% for the phantom. The quantity ε* can be considered a lower bound for the compression errors obtainable by applying any procedure to the noisy blurred image with any noise level.

The matrix A which performs the blur is a 2-level circulant matrix generated by a positive space invariant band limited **PSF** with a bandwidth τ=8, normalized in such a way that the sum of the elements is equal to 1. For the astronomical image we consider a motion-type **PSF**, which simulates the one corresponding to a ground-based telescope, represented by the following mask:

$$m_{i,j} = \exp(-\alpha\ (i+j)^2 - \beta\ (i-j)^2),\ -\tau \le i,j \le \tau, \quad \alpha=0.19,\ \beta=0.17.$$

For the image of medical interest we consider a Gaussian **PSF** represented by the following mask:

$$m_{i,j} = \exp(-\alpha\ i^2 - \beta\ j^2),\ -\tau \le i,j \le \tau, \quad \alpha = \beta = 0.1.$$

For each problem, two noisy blurred images $\boldsymbol{b}$ (one with a small noise level and one with a large noise level) are generated from the vector $\boldsymbol{b}^* = A\boldsymbol{x}^*$ by adding white Gaussian noises of different variance. The obtained noise levels are η=5.6% and η=28% for the satellite, η=4% and η=21% for the phantom. Figure 1 shows the original images X^*, the compressed 10-rank approximations W^*H^* and the noisy blurred images with the large noise level.

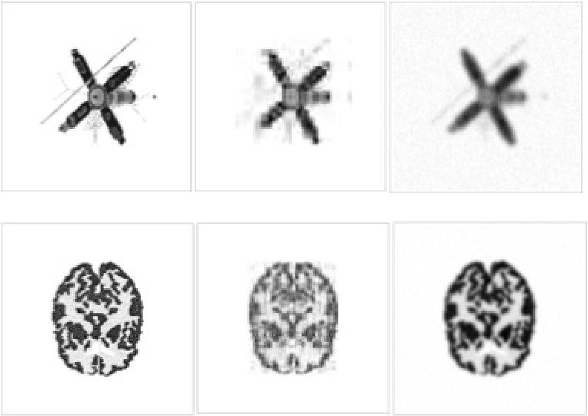

Figure 1. The original image X^* (left), the compressed 10-rank approximation W^*H^* (middle) and the noisy blurred image B with large noise level (right) for the satellite (upper row) and the phantom (lower row).

The regularized compressed images are computed using the procedures **P1**, **P2** and **P3** described in Section 3.2. In the case of procedure **P1**, the denoising is performed in the first phase and this suggests to use in this phase a small value for θ, so we set $\theta = 10^{-4}$. In the case of procedure **P2** two possibly different tolerances could be considered for the inner and the outer steps. Anyway, due to the alternating nature of the procedure, a tight convergence is not required and large tolerance values can be set. The same value $\theta = 10^{-2}$ for both the steps has been found efficacious. Also for the second phase of procedure **P3**, there is no need to fix a too small tolerance value so we set $\theta = 10^{-2}$.

Procedure **P1** computes the regularized image X_{reg} and the factorization $\mathrm{W}^{(1)}\mathrm{H}^{(1)} \approx X_{reg}$ with the errors

$$\varepsilon_{sol} = \frac{\|X^* - X_{reg}\|_F}{\|X^*\|_F}, \qquad \varepsilon^{(1)} = \frac{\|X^* - W^{(1)}H^{(1)}\|_F}{\|X^*\|_F}.$$

Procedure **P2** computes the initial factorization W_0H_0 by an incomplete **NMF** and the factorization $W^{(2)}H^{(2)}$ by solving directly problems (18) with the errors

$$\varepsilon_{fac} = \frac{\|X^* - W_0H_0\|_F}{\|X^*\|_F}, \qquad \varepsilon^{(2)} = \frac{\|X^* - W^{(2)}H^{(2)}\|_F}{\|X^*\|_F}.$$

Procedure **P3** computes the initial factorization W_pH_p using penalized **NMF** and the factorization $W^{(3)}H^{(3)}$ by solving problem (20) with the errors

$$\varepsilon_{fac} = \frac{\|X^* - W_pH_p\|_F}{\|X^*\|_F}, \qquad \varepsilon^{(3)} = \frac{\|X^* - W^{(3)}H^{(3)}\|_F}{\|X^*\|_F}.$$

For procedures **P2** and **P3** the error ε_{fac} is obviously the same for both methods **EM** and **SGP**. Table 1 for the satellite image and Table 2 for the phantom image list

- the errors,
- the total iteration number, given in the form $it = it_{sol} + it_{fac}$,
- the coefficients c_1 of γ_{ft} and c_2 of γ_{kn} in (25)

for both small and large noise levels.

Actually in a practical context, when k and $\log_2 n$ are not too different (as in our experiments) γ_{ft} and γ_{kn} could be assimilated and the sum $C = c_1 + c_2$ would give a simpler idea of the cost.

Table 1. Results of procedures P1, P2 and P3 for satellite.

proc.	met.	small noise level					large noise level				
P1		ε_{sol}	$\varepsilon^{(1)}$	it	c_1	c_2	ε_{sol}	$\varepsilon^{(1)}$	it	c_1	c_2
	EM	27%	35%	131+25	786	50	32%	36%	37+30	222	60
	SGP	27%	34%	34+36	204	72	32%	36%	11+29	66	58
P2		ε_{fac}	$\varepsilon^{(2)}$	it	c_1	c_2	ε_{fac}	$\varepsilon^{(2)}$	it	c_1	c_2
	EM	44%	41%	21+5	126	52	45%	41%	16+5	96	42
	SGP	44%	38%	30+5	180	70	45%	40%	14+5	84	38
P3		ε_{fac}	$\varepsilon^{(3)}$	it	c_1	c_2	ε_{fac}	$\varepsilon^{(3)}$	it	c_1	c_2
	EM	40%	38%	8+20	48	56	41%	40%	5+18	30	46
	SGP	40%	38%	7+20	42	54	41%	39%	5+18	30	46

Table 2. Results of procedures P1, P2 and P3 for phantom.

proc.	met.	small noise level					large noise level				
P1		ε_{sol}	$\varepsilon^{(1)}$	it	c_1	c_2	ε_{sol}	$\varepsilon^{(1)}$	it	c_1	c_2
	EM	33%	42%	185+15	1110	30	35%	43%	29+23	174	46
	SGP	32%	42%	36+20	216	40	35%	43%	11+23	66	46
P2		ε_{fac}	$\varepsilon^{(2)}$	it	c_1	c_2	ε_{fac}	$\varepsilon^{(2)}$	it	c_1	c_2
	EM	50%	48%	14+5	84	38	51%	49%	8+5	48	26
	SGP	50%	47%	15+5	90	40	51%	48%	11+5	66	32
P3		ε_{fac}	$\varepsilon^{(3)}$	it	c_1	c_2	ε_{fac}	$\varepsilon^{(3)}$	it	c_1	c_2
	EM	46%	45%	6+20	36	52	47%	46%	3+18	18	42
	SGP	46%	44%	6+20	36	52	47%	45%	5+18	30	46

By comparing the performances of **EM** and **SGP**, we see that for what concerns the final errors they have a comparable behaviour in nearly all the cases, but their computational costs vary. **SGP**, which has often a better convergence rate, appears to be preferable.

More important are the differences among the procedures. Naturally, one expects that the earlier we get rid of the noise, the best the final result. In fact, procedure **P1**, which acts on an already regularized image, outperforms from the error point of view (but with a higher cost) the two other procedures which act on only partially denoised images. Procedure **P3**, which acts on a better denoised image than **P2**, outperforms it. For what concerns the computational cost, procedure P3 is clearly preferable.

Anyway, in spite of the numerical differences, the compressed images obtained from the final products *WH* are hardly visually distinguishable, as it can be seen in Figure 2 (to be compared with the intermediate image in the bottom row of Figure 1 which refers to the compressed approximation of the original image).

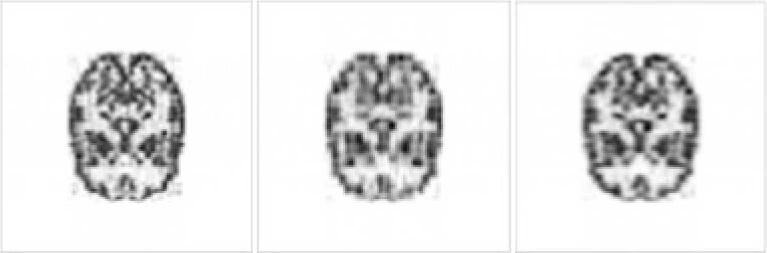

Figure 2. Compressed images from the final products *WH* obtained by procedure **P1** (left), **P2** (middle) and **P3** (right) for the phantom (lower row).

5. Conclusions

In this paper three different procedures have been examined and tested to perform the regularized compression of an image in the form of product of two low rank nonnegative factors. Such an approximation can be obtained by applying first a regularizing step to the given noisy blurred image and then a factorization step (procedure P1), or by combining regularization and factorization steps in an alternating scheme (procedure P2), or by first factorizing the given image with a denoising penalty condition and then applying a deblurring step (procedure P3).

In Figure 3 the trade-off between the final error and the computational cost is shown for the satellite and phantom problems in the case of large noise level. It is self evident that procedure **P1** allows computing a better approximation with a higher cost, but acceptable approximations can be obtained with a significantly reduction of the cost with procedure **P3**.

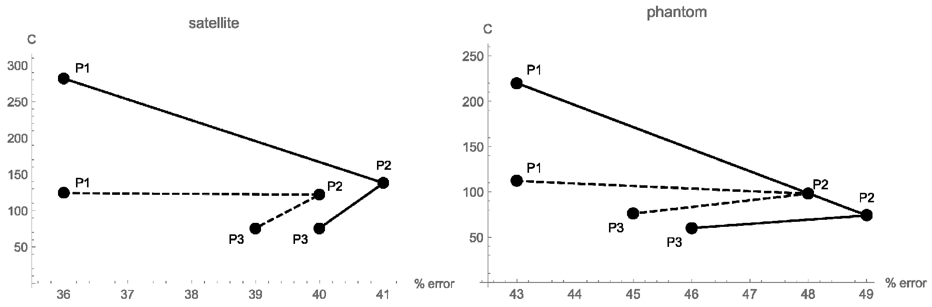

Figure 3. Trade-off between the final errors $\varepsilon^{(i)}$ and the quantity $C = c_1+c_2$ for large noise level, method **EM** (solid line), method **SGP** (dashed line).

References

[1] Pauca, V. P., Piper, J., & Plemmons, R. J. (2006). Nonnegative Matrix Factorization for Spectral Data Analysis. *Linear Algebra Appl.,* vol. 416, (pp. 29-47).

[2] Paatero, P., & Tappert, U. (1994). Positive Matrix Factorization: a non-negative factor model with optimal solution of error estimates of data values. *Environmetrics*, vol. 5, (pp. 111-126).

[3] Kim, J., He, Y., & Park, H. (2014) Algorithms for nonnegative matrix and tensor factorization: an unified view based on block coordinate descent framework. *J. Glob. Optim.*, vol. 58 (pp. 285-319).

[4] Kim, H., & Park, H. (2011). Fast nonnegative matrix factorization: an active-set-like method and comparisons, *SIAM J. on Scientific Computing*, vol. 33, (pp. 3261-3281).

[5] Lawson, C. L., & Hanson, R. J. (1974) *Solving least squares problems*, Prentice-Hall, Englewood Cliffs, N.Y.

[6] Grippo, L., & Sciandrone, M. (2000). On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. *Oper. Res. Lett.,* vol. 26 (pp. 127-136).

[7] Hsieh, C. J., & Dhillon, I. S. (2011). Fast coordinate descent methods with variable selection for non-negative matrix factorization. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* (pp. 1064-1072).

[8] Shepp, L. A., & Vardi, Y. (1982). Maximum likelihood reconstruction for emission tomography. *IEEE Trans. Med. Imag.,* MI-1, (pp. 113-122).

[9] Bonettini, S., Zanella, R., Zanni, L. (2009). A scaled gradient projection method for constrained image deblurring. *Inverse Problems,* 25, 015002.

[10] Golub, G. H., Heath, M., & Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, vol. 21, (pp. 215-223).

[11] Wahba, G. (1977) Practical approximate solutions to linear operator equations when the data are noisy. *SIAM J. Numer. Anal.,* vol. 14, (pp. 651-667).

[12] Favati, P., Lotti, G., Menchi, O., & Romani, F. (2014). Stopping rules for iterative methods in nonnegatively constrained deconvolution. *Applied Numerical Mathematics*, vol. 75, (pp. 154-166).

[13] Favati, P., Lotti, G., Menchi, O., & Romani, F. (2014). Generalized Cross-Validation applied to Conjugate Gradient for discrete ill-posed problems. *Applied Mathematics and Computation*, vol. 243 (pp. 258-268).

[14] Hansen, P. C. (1998) *Rank-Deficient and Discrete Ill-Posed Problems*, SIAM Monographs on Mathematical Modeling and Computation, Philadelphia.

[15] Lee, K. P., Nagy, J. G., & Perrone, L. (2002). Iterative methods for image restoration: a Matlab object oriented approach. `http://www.mathcs.emory.edu/~nagy/RestoreTools`.

[16] Hoffman, E. J., Cutler, P. D., Digby, W. M., & Mazziotta, J. C. (1990). 3-D phantom to simulate cerebral blood flow and metabolic images for PET. *IEEE Trans. Nucl. Sci.*, vol. 37, (pp. 616-620).