New Concepts in Imaging: Optical and Statistical Models
D. Mary, C. Theys and C. Aime (eds)
EAS Publications Series, **59** *(2013) 303–324*

CONSTRAINED MINIMIZATION ALGORITHMS

H. Lantéri[1], C. Theys[1] and C. Richard[1]

Abstract. In this paper, we consider the inverse problem of restoring an unknown signal or image, knowing the transformation suffered by the unknowns. More specifically we deal with transformations described by a linear model linking the unknown signal to an unnoisy version of the data. The measured data are generally corrupted by noise. This aspect of the problem is presented in the introduction for general models. In Section 2, we introduce the linear models, and some examples of linear inverse problems are presented. The specificities of the inverse problems are briefly mentionned and shown on a simple example. In Section 3, we give some information on classical distances or divergences. Indeed, an inverse problem is generally solved by minimizing a discrepancy function (divergence or distance) between the measured data and the model (here linear) of such data. Section 4 deals with the likelihood maximization and with their links with divergences minimization. The physical constraints on the solution are indicated and the Split Gradient Method (SGM) is detailed in Section 5. A constraint on the inferior bound of the solution is introduced at first; the positivity constraint is a particular case of such a constraint. We show how to obtain strictly, the multiplicative form of the algorithms. In a second step, the so-called flux constraint is introduced, and a complete algorithmic form is given. In Section 6 we give some brief information on acceleration method of such algorithms. A conclusion is given in Section 7.

1 Introduction

Inverse problems arise in a variety of important applications in science and industry, such as optical and geophysical imaging, medical diagnostic, remote sensing. More generaly such problem occurs when the measured quantities are not directly the quantities of interest (parameters). In such applications, the goal is to estimate the unknown parameters, given the data. More precisely, denoting y the measured

[1] Laboratoire Lagrange, UMR 7293, Université de Nice Sophia Antipolis, CNRS, Observatoire de la Côte d'Azur, Campus Valrose, 06108 Nice Cedex 2, France

DOI: 10.1051/eas/1359014

data (output of a physical system, generaly corrupted by noise), x the input of the system, $m(a,x)$ the model and a the internal parameters of the model ($m(.,.)$ is a known function), four cases must be considered:

- when y, $m(.,.)$ and x are known, the goal is to identify the optimal values of the internal parameters a of the model $m(a,x)$;
- when y, $m(.,.)$ and a are known, the goal is to found the optimal value of x;
- when x, $m(.,.)$ and a are known, y is easy to compute; it is the direct problem;
- when y, $m(.,.)$ only are known, we can say that we have a "blind inverse problem" which is much more difficult to solve than the previous ones (see for example NMF and blind deconvolution).

To solve such inverse problems we are generally faced with the problem of minimization of a discrepancy function between the noisy data y and the (unnoisy) model $m(a,x)$. The discrepancy function must deal with significant properties from the physical point of view, and must leads to a mathematically tractable minimization problem.

Moreover, to be physically acceptable, the solution is subjected to some specific (physical) constraints that have to be taken into account during the minimization process.

In this paper we are mainly concerned with physical processes described by a linear model. An algorithmic method allowing to deal with the minimization of any stricly convex differentiable discrepancy function is proposed; classical constraints such as positivity and fixed sum (integral) are taken into account.

2 Inverse problems with linear models (Bertero 1989; Bertero *et al.* 1998)

2.1 Linear models

In this case, the model $m\,(a,x)$ is simply described by a linear relation between the unknown (input) signal x and the unnoisy transformed signal $\widetilde{y}$ (output), we simply write:

$$\widetilde{y} = m\,(a,x) = Hx. \tag{2.1}$$

More generally, if H (the parameters of the system) is known, for a given x, we can compute $\widetilde{y}$.

On the other hand we have the experimental data y, that is a noisy version of $\widetilde{y}$. The problem is to find a solution x such that Hx is as close as possible to y. This is generally performed by minimizing a discrepancy function between y and Hx, eventually subject to constraints.

This brief presentation shows that we are typically dealing with an inverse problem (Bertero *et al.* 1998) whose difficulties will be briefly indicated in the following sections. We first give some examples of problems in which the model is described by a linear relation.

2.2 Some examples of linear models

2.2.1 Linear unmixing (Heinz & Chang 2001)

In such problems, the model is described by the relation $\widetilde{y} = Hx$. The experimental data y is a one dimensional optical spectrum sampled at various (equispaced) wavelenghts; these (noisy) observations are obtained for example by the spectroscopic analysis of the light contained in a given pixel of an image. The matrix $[H]$ is formed by the juxtaposition of columns containing the (known) spectra of basis possible component (the endmembers, that is, the elements of a dictionnary), sampled at the same wavelenght as the data. The unknown vector x contains the weights (abundances) corresponding to the endmembers, so that the data vector is described as a weighted sum of the endmembers. The constraints in this problem are the following: the weights must be positive or zero, moreover their sum must be 1 (that is, they express a percentage).

One can think that in order to solve this problem in full generality, a supplementary condition must be that the sum of the components of the data, and the sum of the components of the endmembers must be equal.....

2.2.2 Non negative Matrix Factorization N.M.F. - Hyperspectral data (Lee & Seung 2001; Cichocki *et al.* 2009)

Extending first the previous problem, the model can be described by a matrix equation $\left[\widetilde{Y}\right] = [H]\,[X]$. The matrix $[H]$ is the one described in the previous problem, it contains the "endmembers". The unknowns are organized in a matrix $[X]$, each column of this matrix contains the weights (abundances), so that the column "n" of $\left[\widetilde{Y}\right]$ is modeled as the sum of the endmembers (columns of $[H]$) weighted by the elements of the column "n" of $[X]$.

The experimental data are organized in a matrix $[Y]$; each column of $[Y]$ is an optical spectrum analogous to those considered in the previous problem, they correspond to all the pixels of an image. If the matrix $[H]$ is known, the problem will be a simple succession of "linear unmixing" problems.

The NMF problem becomes much more complicated because the endmembers are not known, so that the matrix $[H]$ is unknown as well as $[X]$.

Roughly speaking, the problem is then: knowing the (noisy) data matrix $[Y]$ described as the product of two matrix $[H]$ and $[X]$, found such two matrices subject to some constraints.

2.2.3 Deconvolution (Andrews & Hunt 1977; Demoment 1989; Bertero *et al.* 2008)

Let us consider the case of images. In the space of continuous functions, the model is described by a first kind Fredholm integral with space invariant kernel. After discretization, the data (noisy blurred image), the point spread function (PSF) and the unknown object are obtained as tables of dimensions (N*N) and the model is described by a discrete convolution between the PSF and the object (that can be easily performed using FFT). However for sake of generality, we adopt a matrix notation, so that the columns (or rows) of the data and of the unknown object tables are organized in stack vectors y and x respectively (lenght N^2), the transformation matrix H is then $(N^2 * N^2)$, moreover, if the kernel of the Fredholm equation is space invariant, H is Block-Toeplitz; note that this is not the case for example in medical imaging where, while we have a linear model, the kernel is no more space invariant and corellatively, the matrix H does not have any specific property.

Let us focus more specifically on the deconvolution problem for astrophysical imagery. In such a case, the kernel of the integral equation is not only space invariant, but also positive and moreover, its integral is equal to 1, so that the convolution (blurring operation) of a positive object of known integral gives a positive image with the same integral; such a convolution acts as a low pass spatial filtering operation. The intensities in the image pixels have been redistributed, while the total intensity in the image is equal to the total intensity in the object.

For the discretized problem, this is analogous to say that each column of the matrix H is of sum 1.

The first constraint of our problem is then: the "solution must be positive or zero", while a second constraint will be "the flux must be maintained". Frequently, this last constraint is not clearly taken into account. One can consider that the deconvolution problem is closely related to the "linear unmixing" problem with however some specific difficulties due to the low pass filtering effect of such convolution.

2.2.4 Blind deconvolution (Ayers & Dainty 1988; Lane 1992)

The blind deconvolution can be considered with respect to the classic deconvolution as an analogous of the NMF problem with respect to the linear unmixing problem. Indeed, the data model boils down to the convolution product of two unknown functions, then, the number of unknown is (two times) higher than the number of data values; the convolution is however commutative, while for NMF, the matrix product is not, moreover the specific problems appearing in classic deconvolution obviously remains. Then, this problem is very hard to solve and it is out of the scope of the present analysis.

2.3 Some generalities on inverse problems

Inverse problems are generally ill-posed problems in the sense of Hadamard; the conditions of Hadamard (1923) for well-posed problems are:

- the solution must "exist"
- the solution must be "unique"
- the solution must be "stable with respect to the measurement errors" (the noise).

If any of these conditions is not fulfilled, the problem is "ill-posed".

While in finite dimensional spaces, the difficulties linked to the existence and uniqueness of the solution could be circumvented, the problem of stability remains because it is a consequence of the ill-conditionning of the matrix H, that is the condition number K of the matrix H (ratio of the maximum singular value to the minimun singular value $K = \frac{\lambda_{Max}}{\lambda_{min}}$) is high.

To clarify this point in a very simple way, let us consider a simple system of two linear equations with two unknowns, illustrated in Figures 1 and 2.

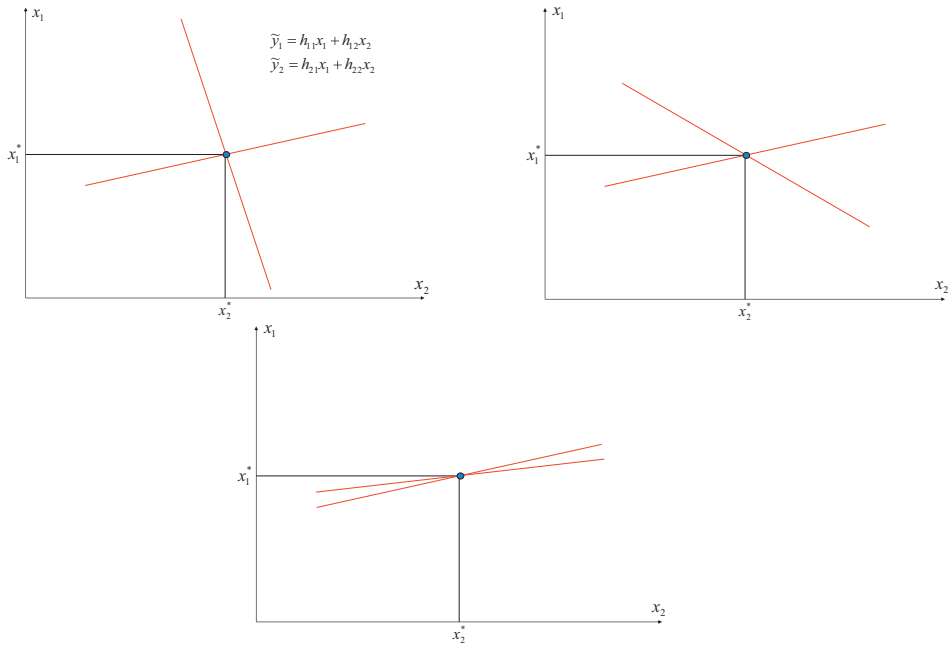

Fig. 1. For a system of two equations with two unknown, three cases are examined, depending on the condition number of the matrix H.

The Figure 1 correspond to the case of an unperturbed system (no noise added). In Figure 1 (upper left), the two lines are almost orthogonal ($K \approx 1$), the system is very well conditionned. If we think for example, to solve the system with an

iterative method operating by successive orthogonal projections on the two lines (1 iteration = 2 projections), it is clear that only a very small number of iterations is necessary to reach an acceptable point (close enough to the solution).

In Figure 1 (upper right), the condition number K has been increased, the two lines are no more orthogonal. Using the iterative method previously described, the iteration number allowing to reach the solution has been increased, but we can expect to reach an acceptable point.

In Figure 1 (lower), the value of K has been strongly increased, the problem is now ill-conditionned, clearly, the solution is always unique, but the necessary number of iterations heavily increases.

To summarize, the only difference between the three cases is an increase of the iteration number and then of computing time when the problem become ill-conditionned.

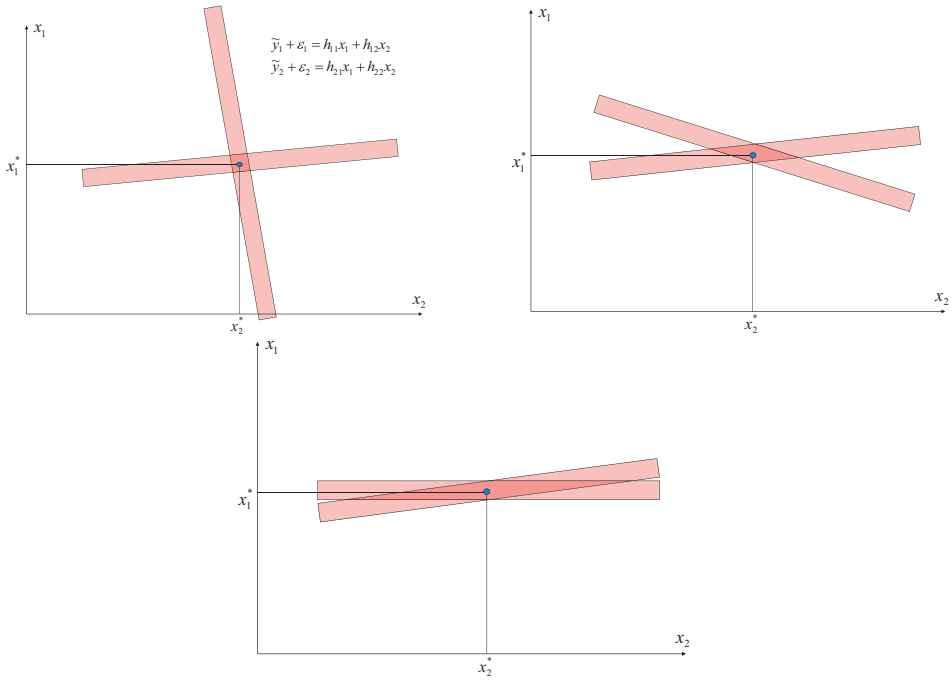

Fig. 2. For a system of two equations with two unknown, three cases are examined, depending on the condition number of the matrix H. A small amount of noise ϵ has been added to the unnoisy data $\widetilde{y}$.

In Figure 2, a small amount of error (noise) has been added to the data.

Depending on the value of the error, the lines remains parallel to themselves, but moves in their respective shaded areas.

Clearly, there will be always one and only one solution that will be located somewhere in the intersection of the two shaded areas.

In Figure 2 (upper left), the solution is close to the one of the initial noiseless problem Figure 1 (upper left) then, a small error on the data will corresponds to a small error on the solution, this behavior is typical of the well-posed problems. In Figure 2 (upper right), the condition number K increases as in Figure 1 (upper right). The solution is always unique and located in the intersection of the shaded areas, but it can be in some cases far from the solution of the initial noiseless problem. In Figure 2 (lower), K has a very large value, the problem is now ill-conditionned and the solution can be very far from the true solution Figure 1 (lower).

This is a simplebut explicit illustration if the difficulties related to the stability of the solution with respect to the measurement errors in ill-posed problems.

3 Distances and divergences (Basseville 1996; Taneja 2005)

To solve the inverse problem, *i.e.* to recover the solution x such that the model $m(a, x)$ is as close as possible to the noisy data y, we must minimize a scalar discrepancy function between y and $m(a, x)$ quantifying the gap between them.

Let p_i and q_i the elements of two data fields p and q, the discrepancy function $D(p, q)$ between the two fields must have the following properties:

1. $D(p, q)$ must be positive if $p \neq q$
2. $D(p, p) = 0$
3. $D(p, q)$ must be convex (strictly) with respect to p and q (at least w.r.t. the field corresponding to the model).

With these properties, $D(p, q)$ is a "divergence". If, moreover the triangular inequality is fullfilled, then $D(p, q)$ is a distance. This last point is not necessary for our purpose. Finally, we consider that generally, such quantity allowing to deal with the whole data fields is the sum of analogous distances (divergences) between corresponding elements of the two fields.

$$D\left(p, q\right) = \sum_i D\left(p_i, q_i\right) \tag{3.1}$$

$$D\left(y, m\left(a, x\right)\right) = \sum_i D\left(y_i, \left\{m\left(a, x\right)\right\}_i\right). \tag{3.2}$$

3.1 Csiszar divergences (Csiszar 1991)

Let $f\left(x\right)$ be a strictly convex function, with $f\left(1\right) = 0$, and for our specific use $f^{'}\left(1\right) = 0$; this last property is very important in our case.

The general class of Csiszar divergences is defined as:

$$C_f\left(p, q\right) = \sum_i q_i f\left(\frac{p_i}{q_i}\right). \tag{3.3}$$

Generally $C_f(p,q) \neq C_f(q,p)$.
This divergence is jointly convex w.r.t. p and q.

3.2 *Divergences founded on convexity measures*

3.2.1 Jensen or Burbea-Rao divergences (Burbea & Rao 1982)

This class of divergences is founded on the classical definition of the convex functions that can be expressed as: let $f(x)$, a strictly convex function, and let p and q two values of the variable, the secant between the points $\{p, f(p)\}$ and $\{q, f(q)\}$ is always superior to the curve between the same points. This is represented in Figure 3 and expressed by the relation (3.4)

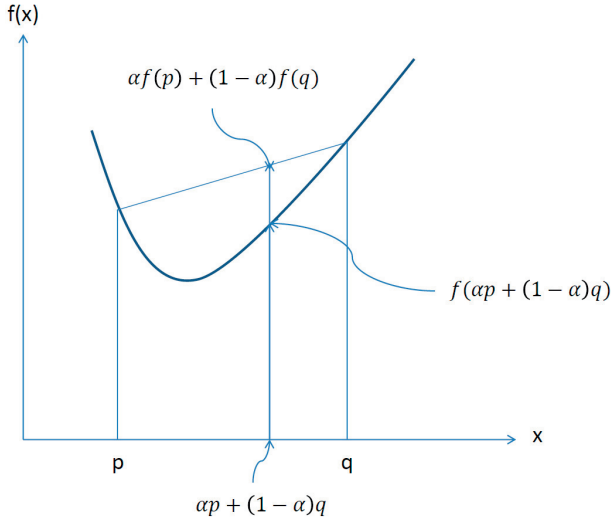

Fig. 3. Strictly convex function.

$$\alpha f(p) + (1-\alpha) f(q) - f[\alpha p + (1-\alpha) q] \geq 0. \tag{3.4}$$

The divergence is then:

$$J_f(p,q) = \sum_i \{\alpha f(p_i) + (1-\alpha) f(q_i) - f[\alpha p_i + (1-\alpha) q_i]\}. \tag{3.5}$$

Note that the convexity of the basis function $f(x)$ does not ensure the conxexity of the corresponding divergence.

3.2.2 Bregman divergences (Bregman 1967)

These divergences are founded on another property of convex functions:
a (strictly) convex function is always greater than any tangent line, that is to the

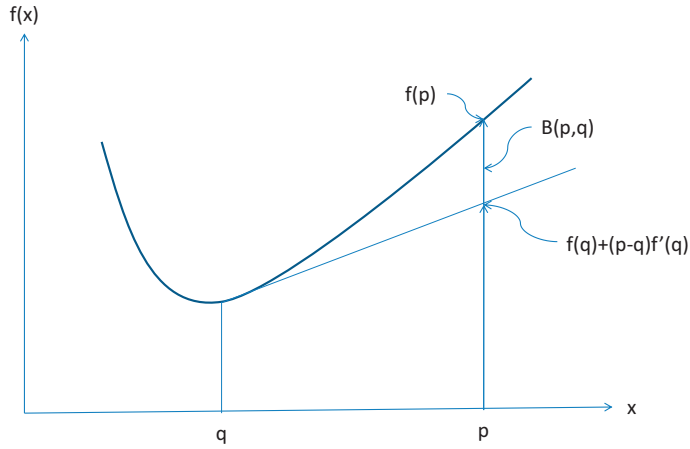

Fig. 4. First order Taylor expansion of a strictly convex function.

first order Taylor expansion of the function; this is represented in Figure 4 and expressed by the relation (3.6)

$$f(p) - f(q) - (p-q) f'(q) \geq 0. \tag{3.6}$$

The Bregman divergence is then:

$$B_f(p,q) = \sum_i \left\{ f(p_i) - f(q_i) - (p_i - q_i) f'(q_i) \right\}. \tag{3.7}$$

Note that this divergence is always convex w.r.t. p, but its convexity w.r.t. q depends on the function f.

This classification of divergences is artificial because it is founded on their constructive method only. A Jensen or Bregman divergence can also be a Csiszar divergence. Moreover, in this brief presentation, we do not consider the extensions or generalization of these divergences, but it is important to know that they exist and could be used as well. Then, at this point it is clear that there are many ways to quantify the discrepancy between two data fields; the question is then: how to choose a "good", that is a "significant" divergence or distance. A partial answer is given by the Maximum Likelihood principle.

4 Maximum likelihood solutions (Taupin 1988)

In this case, we take into account the statistical properties of the noise corrupting the data. We consider that we know the analytical expression of the likelihood that is of the conditional probability law $p(y/x)$, and we want to obtain the value of x corresponding to the maximum of this law.

In each pixel the noisy data y_i depends on the model $[m(a,x)]_i$ which is the mean value; moreover we assume that the noise realizations in the different pixels

are independent. In what follows, the internal parameters a of the model are known, so they are omitted in our notations, $[m(a,x)]_i = [m(x)]_i$, then we have:

$$p(y|m(x)) = \prod_i p(y_i|\{m(x)\}_i) \tag{4.1}$$

and

$$\max_x [p(y|m(x)] \equiv \min_x \left[-\ln \prod_i p(y_i|\{m(x)\}_i) \right]. \tag{4.2}$$

The solution x is obtained as:

$$x = \arg\min \sum_i -\ln [p(y_i|\{m(x)\}_i)]. \tag{4.3}$$

Two cases are generally exhibited in the literature corresponding to physical situations, the zero mean Gaussian additive noise and the Poisson process. We now examine these two cases and we show the relations with the divergences minimization problem.

4.1 Gaussian additive noise case

The likelihood is given by:

$$p(y|x) = p(y|m(x)) \approx \prod_i \exp - \frac{[y_i - \{m(x)\}_i]^2}{\sigma_i^2} \tag{4.4}$$

where σ_i^2 is the noise variance in the pixel i. This leads to an objective function which is the Euclidean distance between y and $m(x)$ in a space weighted by the variances:

$$J(x) = -\ln[p(y|m(x))] \approx \frac{1}{2} \sum_i \frac{[y_i - \{m(x)\}_i]^2}{\sigma_i^2}. \tag{4.5}$$

If the variance is not known or if the variance is identical for all the pixels, we obtain the pure Euclidean distance:

$$J(x) \approx \frac{1}{2} \sum_i [y_i - \{m(x)\}_i]^2. \tag{4.6}$$

One can observe that such a distance is defined for any value of x even if $m(x) \leq 0$.

4.2 Poisson noise case

The likelihood is given by:

$$p(y|x) = p(y|m(x)) = \prod_i \frac{[\{m(x)\}_i]^{y_i}}{y_i!} \exp[-\{m(x)\}_i] \tag{4.7}$$

$$J(x) = -\ln[p(y|m(x))] = \sum_i \{m(x)\}_i + \ln y_i! + y_i \ln \frac{y_i}{\{m(x)\}_i}. \tag{4.8}$$

Which is equivalent to:

$$J(x) = \sum_i \{m(x)\}_i - y_i + y_i \ln \frac{y_i}{\{m(x)\}_i}. \tag{4.9}$$

This expression is the Kullback-Leibler divergence (Kullback & Leibler 1951), adapted to data fields that are not necessarily probability laws. On the contrary to the Euclidean distance, one can note that the K.L. divergence is not defined if $m(x) \leq 0$. In the case of our linear model $x > 0 \Rightarrow m(x) = Hx > 0$. When the positivity constraint is required, the constraints domain is entirely contained in the domain of the objective function $J(x)$. Then if the solution is searched for in the constraints domain, the minimization can be performed. It is one of the reasons which leads to use an interior points algorithmic method. In such an iterative method, the successive estimates are feasibles solutions *i.e.* they fulfill all the constraints. We propose now a minimization method dealing with strictly convex differentiable functionnals, subject to a constraint on the inferior bound of the solution. The positivity constraint will appear as a particular case.

5 The Split Gradient Method (SGM)

This iterative method has been developped initially in the context of the deconvolution problem with non negativity constraint (Lanteri *et al.* 2001, 2002a,b). The multiplicative form of the algorithms is an obvious byproduct. It can be easily extended to regularized functionnals. The method is founded on the Karush-Kuhn-Tucker (KKT) conditions (Bertsekas 1995). We first recall these conditions. A simple example with only one variable clarifies this point.

5.1 Karush-Kuhn-Tucker conditions for inequality constraints

We denote $J_1(x)$, the "data consistency" term, $J_2(x)$, the "regularization" term and $\gamma \geq 0$, the regularization factor. The problem is to minimize w.r.t. x, the strictly convex differentiable functionnal: $J(x, \gamma) = J_1(x) + \gamma J_2(x)$.

The constraints are: $x_i \geq m_i \geq 0 \ \ \forall i \quad \Rightarrow \quad x_i - m_i \geq 0 \ \ \forall i$.

In what follows, the parameter γ will be omitted for sake of clarity.

Let us denote λ the Lagrange multiplier vector, and $\langle .,. \rangle$ the classical inner product.

The Lagrange function writes:

$$L(x, \lambda) = J(x) - \langle \lambda, (x - m \rangle). \tag{5.1}$$

The KKT conditions writes: at the solution (x^*, λ^*)

$$\nabla_x L(x^*, \lambda^*) = 0 \quad \Rightarrow \quad \lambda^* = \nabla_x J(x^*) \tag{5.2}$$

$$\lambda^* \geq 0 \quad \Rightarrow \quad \nabla_x J\left(x^*\right) \geq 0 \tag{5.3}$$

$$\left\langle \lambda^*, \left(x^* - m\right)\right\rangle = 0 \;\Rightarrow\; \left\langle \nabla_x J\left(x^*\right), \left(x^* - m\right)\right\rangle = 0. \tag{5.4}$$

This last condition must be understood as:

$$\left[\nabla_x J\left(x^*\right)\right]_i \left(x_i^* - m_i\right) = 0 \quad \forall i. \tag{5.5}$$

Indeed, because $x_i^* - m_i \geq 0\ \forall i$, and $\left[\nabla_x J\left(x^*\right)\right]_i \geq 0\ \forall i$, the inner product will be zero if and only if all the terms of the inner product are separately 0.

The Split Gradient Method is founded precisely on this condition.

The KKT conditions for inequality constraints can be understood easily on a simple example for a function of one variable $f(x)$ with an inferior bound constraint $x \geq m$.

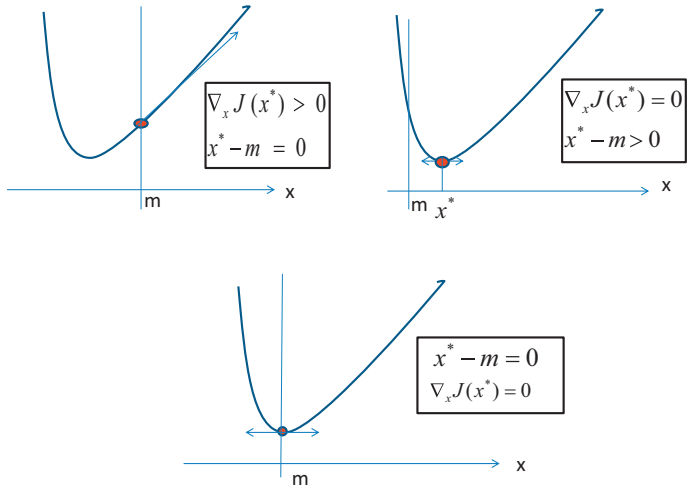

Fig. 5. Illustration of KKT conditions for non negativity constraint in the one dimensional case.

Let us consider the case represented in Figure 5 (upper right).

The minimum of the function is clearly reached for x such that $f'(x) = 0$. The solution is the same to the one of the unconstrained problem; in the case of a function of several variables, the solution will be reached when $\left[\nabla_x f\left(x\right)\right]_i = 0$, for the corresponding components i.

Then, for such indexes, $\left(x_i - m_i\right)\left[\nabla_x f\left(x\right)\right]_i = 0$ because $\left[\nabla_x f\left(x\right)\right]_i = 0$.

In Figure 5 (upper left), the solution is on the constraint $x - m = 0$. At this point, we have $\lambda = f'(x) > 0$. In a multi variables case the equivalent condition will be: if a component i is on the constraint $x_i - m_i = 0$, we will have $\lambda_i = \left[\nabla_x f\left(x\right)\right]_i > 0$, so that **for each component on the constraint**, we will have $\left(x_i - m_i\right)\left[\nabla f\left(x\right)\right]_i = 0$.

In Figure 5 (lower), the minimum of the function is exactly on the constraint, so that we have simultaneously $x - m = 0$ and $f^{'}(x) = 0$, then obviously, their product is zero. In a multi variables case the equivalent condition will be: if a component i is on the constraint $x_i - m_i = 0$, and if moreover for the same components, we have $\lambda_i = [\nabla_x f(x)]_i = 0$, for such components we will have $(x_i - m_i)[\nabla_x f(x)]_i = 0$.

Then for each component of the solution, the KKT condition expresses as: $(x_i - m_i)[\nabla_x f(x)]_i = 0$.

5.2 Principle of the Split Gradient Method

The problem is set as: let $\gamma \geq 0$ and y the noisy data, found

$$x = \arg\ \min\ J(x, \gamma) = J_1(x) + \gamma J_2(x). \tag{5.6}$$

Subject to the constraint

$$0 \leq m_i \leq x_i \quad \forall i. \tag{5.7}$$

Moreover, in the particular case $m_i = 0\ \forall i$, we will introduce a supplementary equality constraint:

$$\sum_i x_i = \sum_i y_i. \tag{5.8}$$

In a first step the equality constraint is not considered; it will be introduced later. Considering now that for convex differentiable functionnals such as $J(x, \gamma) \equiv J(x)$, the negative gradient is a descent direction, we want to solve w.r.t. x an equation of the form:

$$[-\nabla_x J(x^*)]_i (x_i^* - m_i) = 0 \quad \forall i. \tag{5.9}$$

Note that the multiplication of this equation by a positive term do not change solution.

Then, an iterative algorithm can be writen in the form:

$$x_i^{k+1} = x_i^k + \alpha_i^k \left(x_i^k - m_i\right) \left[-\nabla_x J\left(x^k\right)\right]_i. \tag{5.10}$$

In this algorithm, α_i^k is a positive descent step that must be computed to ensure the convergence of the algorithm. Moreover the form of the algorithm implies that at each iterative step, we must ensure that $x_i^k - m_i \geq 0$. This last point is of major importance in SGM.

The negative gradient is now written as the difference between two positive quantities $U\left(x^k\right)$ and $V\left(x^k\right)$:

$$-\nabla_x J\left(x^k\right) = U\left(x^k\right) - V\left(x^k\right). \tag{5.11}$$

Obviously, such a decomposition is not unique, indeed a constant term can be added and subtracted to the gradient, leading to shifted values of U and V, with the only condition that the shifted values remains positive.

We then propose to modify the algorithm as follows:

$$x_i^{k+1} = x_i^k + \alpha_i^k \left(x_i^k - m_i\right) \frac{1}{\left[V\left(x^k\right)\right]_i} \left[\underbrace{U\left(x^k\right) - V\left(x^k\right)}_{-\nabla J(x^k)}\right]_i \cdot \tag{5.12}$$

In the rest of the paper, we will use for sake of clarity, the notations: $\left[U\left(x^k\right)\right]_i = U_i^k$ and $\left[V\left(x^k\right)\right]_i = V_i^k$.

We can observe that the descent property is maintained even if the descent direction is changed.

The starting iterate will be $x_i^0 \geq m_i, \quad \forall i$.

The first step of the method is to compute for each component of the solution vector, the maximal step size ensuring $x_i^{k+1} \geq m_i \quad \forall i$, knowing that $x_i^k \geq m_i \quad \forall i$.

Obviously, such restriction on the step size is only necessary for the indexes i for which $\left[\nabla J\left(x\right)\right]_i \geq 0$.

This leads to:

$$\alpha_i^k \leq \frac{V_i^k}{V_i^k - U_i^k} \cdot \tag{5.13}$$

Then, at the iteration "k" the maximal step size allowing to fulfill the inferior bound constraint for all components, will be:

$$\alpha_{Max}^k = \min_i \left[\alpha_i^k\right]. \tag{5.14}$$

We note that $\alpha_{Max}^k \geq 1$.

As a consequence, with a stepsize equal to 1 the inferior bound constraint is always fulfilled. The proposed algorithm can then be written in matrix form:

$$x^{k+1} = x^k + \alpha_c^k \, diag\left[x_i^k - m_i\right] diag\left[\frac{1}{V_i^k}\right] \underbrace{\left(U^k - V^k\right)}_{-\nabla J^k} \cdot \tag{5.15}$$

It is a descent algorithm of scaled gradient type, that is of the general form:

$$x^{k+1} = x^k + \alpha_c^k d^k. \tag{5.16}$$

The descent direction is:

$$d^k = diag\left[x_i^k - m_i\right] diag\left[\frac{1}{V_i^k}\right] \underbrace{\left(U^k - V^k\right)}_{-\nabla J^k} \cdot \tag{5.17}$$

The descent stepsize α_c^k must be computed on the range $\left[0, \alpha_{Max}^k\right]$ to ensure the convergence of the algorithm. However, if we use a stepsize equal to 1 $\forall k$, we obtain a very attractive simple "quasi multiplicative" form, whose convergence is not demonstrated in full generality, but only in some specific cases:

$$x^{k+1} = m + diag\left[x^k - m\right] \frac{U^k}{V^k} \cdot \tag{5.18}$$

For a non negativity constraint ($m_i = 0\ \forall i$), the classical multiplicative form is immediately obtained:

$$x^{k+1} = diag\left[x^k\right] \frac{U^k}{V^k}. \tag{5.19}$$

In the two last equations, the ratio $\frac{U^k}{V^k}$ of the vectors U^k and V^k is performed component wise. With this simplified form, we can recover two classical algorithms: ISRA (Daube-Witherspoon *et al.* 1986) and RLA (Richardson 1972; Lucy 1974) corresponding respectively to the hypothesis of a Gaussian, zero mean additive noise, and to a Poisson noise process.

5.3 Examples with non negativity constraint

5.3.1 Gaussian additive noise case - Least squares

As previously indicated in Equation (4.6), the objective function writes:

$$J(x) = \frac{1}{2}\left\|y - Hx\right\|^2 \tag{5.20}$$

$$-\nabla J(x) = H^T y - H^T H x. \tag{5.21}$$

A decomposition can be:

$$U = H^T y; \quad V = H^T H x \tag{5.22}$$

Then the algorithm with non negativity constraint, in the non-relaxed form writes:

$$x_i^{k+1} = x_i^k \frac{\left(H^T y\right)_i}{\left(H^T H x^k\right)_i}. \tag{5.23}$$

This is the classical Image Space Reconstruction Algorithm (ISRA) whose convergence has been demonstrated by De Pierro (1987).

If some of the components of $V = H^T y$ is negative, we can add to all the components of U and V, the quantity $-\min\left(H^T y\right) + \epsilon$, so that the shifted values become positive.

5.3.2 Poisson noise case - Kullback-Leibler divergence

As previously indicated in Equation (4.9), the objective function writes:

$$J(x) = \sum_i y_i \ln \frac{y_i}{(Hx)_i} + (Hx)_i - y_i \tag{5.24}$$

$$-\nabla J(x) = H^T\left(\frac{y}{Hx} - \mathbf{1}\right). \tag{5.25}$$

In this equation the ratio of two vectors is performed component wise; the result of the operation is a vector. A decomposition can be:

$$U = H^T \frac{y}{Hx}; \quad V = H^T \mathbf{1}. \tag{5.26}$$

Then the algorithm with non negativity constraint, in the non-relaxed form writes:

$$x_i^{k+1} = x_i^k \frac{\left(H^T \frac{y}{Hx^k}\right)_i}{\left(H^T \mathbf{1}\right)_i} = x_i^k \frac{\left(H^T \frac{y}{Hx^k}\right)_i}{a_i}. \tag{5.27}$$

This is the classical E.M. (Dempster *et al.* 1977), Richardson-Lucy algorithm.

Some remarks then occur:

1. In the previous equation we have introduced the notation: $\left(H^T \mathbf{1}\right)_i = a_i$, howe-ver, in many cases for example in deconvolution problem with a convolution kernel normalized to "1", all the columns of H are of sum 1, that is $a_i = 1 \ \forall i$. Unfortunately, an oversimplified expression of the algorithm in which a_i is omitted, frequently appears; this can be a source of errors.

2. The algorithm of Richardson-Lucy with a kernel normalized to "1", have the "magic" and unwanted property to be flux conservative, that is $\sum_i x_i^k = \sum_i y_i \ \forall k$; this property does not exist with ISRA.

An interesting question is: why?

The answer lies in the particular expression of the K.L. divergence and in the associated properties.

5.4 Flux (intensity) conservation constraint (Lanteri et al. 2009, 2010)

We propose now to introduce a supplementary equality constraint in order to take into account the so called flux constraint or fixed sum constraint. While the method can be applied to the problem adressed in the previous section with a constant inferior bound constraint (which is typical of deconvolution problems), for sake of simplicity, we restrict the presentation to the case of a non negativity constraint.

The equality constraint writes:

$$\sum_i x_i = \sum_i y_i. \tag{5.28}$$

Moreover, because we want to remain in the class of interior points methods, such a constraint must be fulfilled at each iteration, that is:

$$\sum_i x_i^k = \sum_i y_i \quad \forall k. \tag{5.29}$$

The two previous relations expressing a sum constraint are typical of the deconvolution problem. In problems such as the linear unmixing one, we have to simply replace $\sum_i y_i$ by 1, without changing anything else in what follows.

The basic idea to take into account this constraint is to use the following procedure:

- Introduce the variable change:

$$x_i = \frac{u_i}{\sum_m u_m} \sum_m y_m. \tag{5.30}$$

- Proceed to a minimization w.r.t. the new variable u, subject to non negativity constraint only.
- Go back (correctly) to the initial variables x.

To minimize w.r.t. the new variable u, subject to non negativity constraint, we use the SGM previously described.

However, a fundamental question arises first: if $J(x)$ is convex w.r.t. x, did the function $\widetilde{J}(u)$ transformed function of $J(x)$ is still convex w.r.t. u?

The answer may be as follows: if during the iterative minimization process w.r.t. u, we are able to maintain $\sum_i u_i^k = Cst \ \ \forall k$, then the convexity w.r.t. u is ensured.

Moreover, we show that this property will allow us to go back "correctly" to the initial variables x.

To apply SGM, we compute the gradient of $\widetilde{J}(u)$ w.r.t. u

$$\frac{\partial \widetilde{J}(u)}{\partial u_j} = \sum_i \frac{\partial J}{\partial x_i} \frac{\partial x_i}{\partial u_j}. \tag{5.31}$$

We then obtain after some simple but tedious algebra:

$$-\frac{\partial \widetilde{J}(u)}{\partial u_j} \approx \left(-\frac{\partial J}{\partial x_j}\right) - \sum_i \frac{u_i}{\sum_m u_m} \left(-\frac{\partial J}{\partial x_i}\right). \tag{5.32}$$

We can now use SGM to minimize $\widetilde{J}(u)$ w.r.t. u with the non negativity constraint only, but we want also that $\sum_i u_i^{k+1} = \sum_i u_i^k \ \forall k$.

To reach such an objective, at first sight, we can choose:

$$\begin{aligned} U_j &= -\frac{\partial J}{\partial x_j} = \left(-\frac{\partial J}{\partial x}\right)_j \\ V_j &= \sum_i \frac{u_i}{\sum_m u_m} \left(-\frac{\partial J}{\partial x_i}\right) = \sum_i \frac{u_i}{\sum_m u_m} \left(-\frac{\partial J}{\partial x}\right)_i. \end{aligned} \tag{5.33}$$

However, with such a choice, we cannot ensure that U_j and V_j are positive.

To have this property which is necessary in S.G.M., we will choose:

$$U_j = \left(-\frac{\partial J}{\partial x}\right)_j - \min\left(-\frac{\partial J}{\partial x}\right) + \epsilon \tag{5.34}$$

$$V_j = \sum_i \frac{u_i}{\sum_m u_m} \left(-\frac{\partial J}{\partial x}\right)_i - \min\left(-\frac{\partial J}{\partial x}\right) + \epsilon \tag{5.35}$$

One can also write:

$$V_j = \sum_i \frac{u_i}{\sum_m u_m} \left[\left(-\frac{\partial J}{\partial x}\right)_i - \min\left(-\frac{\partial J}{\partial x}\right) + \epsilon\right]. \tag{5.36}$$

Obviously, the shift $-\min\left(-\frac{\partial J}{\partial x}\right) + \epsilon$ does not change the gradient, but now, we are sure that U_j and V_j are positive. Let us note that V_j is in fact constant and independent of the index j.

We can now apply SGM to obtain the relaxed algorithm:

$$u_j^{k+1} = u_j^k + \alpha^k u_j^k \left(\frac{\left(-\frac{\partial J}{\partial x^k}\right)_j - \min\left(-\frac{\partial J}{\partial x^k}\right) + \epsilon}{\sum_i \frac{u_i^k}{\sum_m u_m^k}\left[\left(-\frac{\partial J}{\partial x^k}\right)_i - \min\left(-\frac{\partial J}{\partial x^k}\right) + \epsilon\right]} - 1 \right). \tag{5.37}$$

The step size α^k is obviously computed as indicated in Section 5.2.

In the non relaxed case, that is , with $\alpha^k = 1\ \forall k$, we have:

$$u_j^{k+1} = u_j^k \frac{\left(-\frac{\partial J}{\partial x^k}\right)_j - \min\left(-\frac{\partial J}{\partial x^k}\right) + \epsilon}{\sum_i \frac{u_i^k}{\sum_m u_m^k}\left[\left(-\frac{\partial J}{\partial x^k}\right)_i - \min\left(-\frac{\partial J}{\partial x^k}\right) + \epsilon\right]}. \tag{5.38}$$

Clearly, with such form of the algorithm, relaxed or non-relaxed, we will have:

$$\sum_j u_j^{k+1} = \sum_j u_j^k. \tag{5.39}$$

Then, during the iterative process, the solution u^k is positive and remains in the convexity domain of the objective function $\widetilde{J}(u)$. Moreover the flux conservation property of the previous algorithms (5-37, 5-38) allows us to turn back "correctly" to the initial variables x. Indeed, multiplying the two members of these algorithms by $\frac{\sum_m y_m}{\sum_j u_j^{k+1}} = \frac{\sum_m y_m}{\sum_j u_j^k}$, and taking into account the change of variables (5-30), the final algorithm is obtained in the relaxed case as:

Let $x^0 = Cst \geq 0$ such that $\sum_i x_i^0 = \sum_i y_i$,

$$x_j^{k+1} = x_j^k + \alpha^k x_j^k \left(\frac{\left(-\frac{\partial J}{\partial x^k}\right)_j - \min\left(-\frac{\partial J}{\partial x^k}\right) + \epsilon}{\sum_i \frac{x_i^k}{\sum_m y_m}\left[\left(-\frac{\partial J}{\partial x^k}\right)_i - \min\left(-\frac{\partial J}{\partial x^k}\right) + \epsilon\right]} - 1 \right). \tag{5.40}$$

Let us observe that with such a relaxed algorithm, we obtain:

$$\sum_i x_i^{k+1} = (1 - \alpha^k) \sum_i x_i^k + \alpha^k \sum_i y_i. \tag{5.41}$$

So that, the flux conservation is related to the properties of the initial estimate, that is $\sum_i x_i^0 = \sum_i y_i$.

In the non relaxed case, that is, with $\alpha^k = 1 \forall k$, we obtain:

$$x_j^{k+1} = x_j^k \frac{\left(-\frac{\partial J}{\partial x^k}\right)_j - \min\left(-\frac{\partial J}{\partial x^k}\right) + \epsilon}{\sum_i x_i^k \left[\left(-\frac{\partial J}{\partial x^k}\right)_i - \min\left(-\frac{\partial J}{\partial x^k}\right) + \epsilon\right]} \sum_m y_m. \tag{5.42}$$

One can easily check that $x_i^{k+1} \geq 0$ if $x_i^k \geq 0 \ \ \forall k$, and that $\sum_i x_i^{k+1} = \sum_i y_i \ \forall k$ even if $\sum_i x_i^k \neq \sum_i y_i$.

This is basically different of the property of the non-relaxed algorithm.

Unfortunately, to our experience, such beautifull non relaxed algorithm does not converge, and the relaxed version must always be used. The corollary remark is that the only effective property concerning the flux constraint will be:

$$\sum_i x_i^k = \sum_i x_i^0. \tag{5.43}$$

All the algorithms founded on SGM are sometimes considered as having a slow convergence rate. In the relaxed form, the stepsize computation allows to ensure the convergence and moreover to (slightly) modify the convergence speed. Then we briefly indicate in the following section the general rules of the acceleration methods proposed in the literature.

6 Acceleration methods (Biggs *et al.* 1997; Nesterov 1983; Beck *et al.* 2010)

6.1 Principle of the method

Considering that we have a basis convergent algorithm analogous to (5.40), written in the form:

$$x^{k+1} = F(x^k). \tag{6.1}$$

Remember that in such an algorithm, the solution x^{k+1} is at each step non negative and of fixed sum if x^k is non negative and of fixed sum.

The general form of the acceleration methods proposed in the litterature could be summarized as follows:

1. Given the initial estimate x^0 fulfilling all the constraint, compute x^1 (which obviously fulfill all the constraints).

2. Knowing x^k and x^{k-1}, proceed to a linear extrapolation step to obtain the prediction $\widehat{x}^{k+1}$ as:

$$\widehat{x}^{k+1} = x^k + \delta^k \left(x^k - x^{k-1}\right) \tag{6.2}$$

   where the extrapolation step size δ^k is positive or zero $\forall k$.

   Two expressions allowing to obtain this stepsize are given in (Biggs *et al.* 1997; Nesterov 1983; Beck *et al.* 2010); however some supplementary restrictions on this stepsize are necessary as indicated in the comments.

3. Proceed to an iteration of the basis algorithm:

$$x^{k+1} = F(\widehat{x}^{k+1}). \tag{6.3}$$

6.2 Comments

All the difficulties are in the choice of the extrapolation step size, indeed:

- The extrapolated solution $\widehat{x}^{k+1}$ must be a non negative solution.

  Depending on the choice of δ^k, some components of $\widehat{x}^{k+1}$ can become negative, this is not allowed; if one think to project orthogonally $\widehat{x}^{k+1}$ on the space of non negative vectors, then, the flux constraint is not fulfilled; as a conclusion, the extrapolation step, must lead to $\widehat{x}^{k+1} \geq 0$. Then, due to the linearity of the extrapolation step, $\widehat{x}^{k+1}$ will fulfill the flux constraint.

  To fulfill the non negativity constraint on $\widehat{x}^{k+1}$, some restrictions of the extrapolation step size must be introduced.

- Even if such restrictions are taken into account, the algorithm can be non-monotonic, that is, the objective function can increase locally. This could be a source of problems.

  The solution generally proposed is simply to remove the extrapolation step when this happens.

- If the extrapolation is too strong, the accelerated algorithm may even diverge.

Then, clearly, the main problem is in the value of the extrapolation step size. Even if several methods are proposed in the literature to compute such a step size, as far we know, the convergence of accelerated algorithms is not clearly demonstrated and remain an open problem.

7 Conclusion

In the present work, we analyze mainly the inverse problems in which the overall effect of the physical system corresponds to a linear transformation of the input signal. The discrepancy between the experimental noisy data and the linear model must be quantified. Several classes of divergences or distances are then proposed as discrepancy functions. The problem is then to recover the unknown signal by minimization of the adequate divergence, subject to physical constraints.

The main point of this presentation is the Split Gradient Method. When this method has been elaborated, the objective was to recover, using classical optimization ideas, several algorithms that have been proposed in the field of image restoration or deconvolution. The main constraint introduced in these problems was the non negativity constraint. More generally such constraint has been extended to an inferior bound constraint.

In a second step, we have taken into account explicitely the flux conservation or the fixed sum constraint. The corresponding algorithms have been exhibited in the context of the SGM. These algorithms have been applied successfully in the fields of linear unmixing, NMF and deconvolution. Finally, the acceleration method of such algorithms is considered and briefly discussed at the end of the paper.

References

Bertero, M., 1989, Adv. Electr. Elect. Phys., 75, 1

Bertero, M., & Boccacci, P., 1998, Introduction to inverse problems in imaging (IOP Publishing)

Heinz, D.C., & Chang, C.I., 2001, IEEE. Trans. G.R.S, 39, 529

Lee, D.D., & Seung, H.S., 2000, NIPS

Cichocki, A., Zdunek, R., Phan, A.H., & Amari, S.I., 2009, Non negative matrix and tensor factorization (J. Wiley)

Andrews, H.C., & Hunt, B.R., 1977, Digital Image Restoration (Prentice Hall)

Demoment, G., 1989, IEEE Trans. ASSP, 12, 2024

Bertero, M., Lanteri, H., & Zanni, L., 2008, in Mathematical methods in Biomedical imaging and IMRT (Edizioni della normale, Pisa)

Ayers, G.R., & Dainty, J.C., 1988, Opt. Lett., 13, 428

Lane, R.G., 1992, J. Opt. Soc. Am. A, 9, 1508

Hadamard, J., 1923, Lectures on the Cauchy problem in linear partial differential equations (Yale University Press, New Haven)

Basseville, M., 1996, Information: entropies, divergences et moyennes, Technical Report, 1020, IRISA

Taneja, I.J., 2005, On mean divergences measures, Math. ST

Csiszar, I., 1991, Ann. Statist., 19, 2032

Burbea, J., & Rao, C.R., 1982, IEEE Trans. IT, 28, 489

Bregman, L.M., 1967, URSS Comput. Math. Math. Phys., 7, 200

Taupin, D., 1988, Probabilities, data reduction and error analysis in the physical sciences (Les Editions de Physique)

Kullback, S., & Leibler, R.A., 1951, Annals Math. Statistics, 22, 79

Lanteri, H., Roche, M., Cuevas, O., & Aime, C., 2001, Sig. Proc., 54, 945

Lanteri, H., Roche, M., & Aime, C., 2002, Inv. Probl., 18, 1397

Lanteri, H., Roche, M., Gaucherel, P., & Aime, C., 2002, Sig. Proc., 82, 1481

Bertsekas, D., 1995, Non Linear Programming (Athena Scientific)

Daube-Witherspoon, M.E., & Muehlehnner, 1986, IEEE Trans. Medical Imaging, 5, 61

Richardson, W.H., 1972, J. Opt. Soc. Am., 1, 55

Lucy, L.B., 1974, AJ, 79, 745

De Pierro, A.R., 1985, IEEE Trans. Medical Imaging, 6, 124

Dempster, A.D., Laird, N.M., & Rubin, D.B., 1977, J. Royal Stat. Soc. B, 39, 1

Lanteri, H., Theys, C., Benvenuto, F., & Mary, D., 2009, Gretsi

Lanteri, H., Theys, C., Fevotte, C., & Richard, C., 2010, Eusipco

Biggs, D.S.C., & Andrews, M., 1997, Appl. Optics, 36, 1766

Nesterov, Yu., E., 1983, Soviet Math. Dokl, 27, 372

Beck, A., & Teboulle, M., 2010, in Convex Optimization in Signal Processing and Communications, ed. D. Palomar & Y. Eldar (Cambridge University Press), 33