Title	Half-quadratic regularization, preconditioning and applications
Author(s)	Ng, KP
Citation	International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong, China, 2-4 May 2001, p. 32-35
Issued Date	2001
URL	http://hdl.handle.net/10722/46606
Rights	Creative Commons: Attribution 3.0 Hong Kong License

HALF-QUADRATIC REGULARIZATION, PRECONDITIONING AND APPLICATIONS

*Michael Ng**

Department of Mathematics
The University of Hong Kong
Pokfulam Road, Hong Kong

ABSTRACT

In this paper, we address a wide class of image deconvolution or reconstruction situations where a sought image is recovered from degraded observed image. The sought solution is defined to be the minimizer of an objective function combining a data-fidelity term and a edge-preserving, convex regularization term. Our objective is to speed up the calculation of the solution in a wide range of situations. We propose a method applying pertinent preconditioning to an adapted half-quadratic equivalent form of the objective function. The optimal solution is then found using an alternating minimization (AM) scheme. We focus specifically on Huber regularization. We exhibit the possibility get very fast calculations while preserving the edges in the solution. Preliminary numerical results are reported to illustrate the effectiveness of our method.

1. INTRODUCTION

This work addresses a wide class of image reconstruction situations where a sought image $\hat{x} \in R^p$ is recovered from degraded data $y \in R^q$ by minimizing an objective function $J : R^p \to R$ combining a data-fidelity term and a regularization term:

$$\hat{x} = \min_{x \in R^p} J(x) \tag{1}$$

$$J(x) = \|Ax - y\|^2 + \beta \sum_{i=1}^{r} \phi(d_i^T x). \tag{2}$$

In the expression above, $A \in R^{q \times p}$ represents the observation system and $\beta > 0$ is a parameter. The regularization term involves a C^1 convex function $\phi : R \to R$, applied to a set of linear transforms of the image $d_i^T x$ which are typically first or second-order differences between neighboring pixels. Such reconstruction methods are well-known to allow the obtention of high quality image estimates $\hat{x}$ when $\phi(t)$ approaches affine function when $|t| \to \infty$ [4, 1, 6].

However, their use in different image reconstruction and image restoration applications can be practically limited by the numerical cost needed for the calculation of the estimate. The latter is usually calculated using iterative descent algorithms, based on the gradient of J. The calculation speed is mainly limited by the fact that the gradient of J is non-linear and the Hessian matrices are usually ill-conditioned at each iteration. In this paper, we consider a a fast algorithm for the calculation of $\hat{x}$ by using a half-quadratic equivalent form of J combined with pertinent preconditioning.

2. TWO FORMS OF HALF-QUADRATIC REGULARIZATION

Since [10] and [11], numerous algorithms have been proposed where $\hat{x}$ is calculated by minimizing an augmented objective function $G : R^p \times R^r \mapsto R$ which involves an auxiliary variable $s \in R^r$:

$$G(x, s) = \|Ax - y\|^2 + \beta \sum_{i=1}^{r} Q(d_i^T x, s_i) + \beta \sum_{i=1}^{r} \psi(s_i)$$

where for every $s_i \in R$ the function $Q(., s_i) : R \mapsto R$ is quadratic. The equivalence with (1-2) is ensured by the requirement that for every $x \in R^p$,

$$J(x) = \min_{s \in R^r} G(x, s).$$

The advantage of such an equivalent formulation is twofold: for every s fixed, the gradient of $x \mapsto G(x, s)$ is an affine function whereas the non-quadratic part of G, that is $s \mapsto G(x, s)$, is separable in s and hence easy to minimize in a parallel way.

Some researchers have proposed and developed quadratic terms of the form [2, 6, 12, 8]:

$$Q(d_i^T x, s_i) = (d_i^T x)^2 s_i. \tag{3}$$

A slightly different form for Q has been proposed in [11] to perform stochastic optimization for a specific concave regularization:

$$Q(d_i^T x, s_i) = (d_i^T x - s_i)^2. \tag{4}$$

Several authors generalized this approach [7, 1]. Besides, the numerical effects relevant to these formulations have never been examined.

Recall that the numerical efficiency of the minimization of G is tightly connected with the structure of its Hessian at each iteration. We observe that under (3), the Hessian of $G(., s)$ is

$$2A^T A + \beta D^T \text{diag}(s) D \tag{5}$$

where D denotes the $p \times r$ matrix yielding $[Dx]_i = d_i x$ for all i and x and $\text{diag}(s)$ is a diagonal matrix with diagonal elements s_i. A drawback of this formulation is that the conditioning of $G(., s)$ depends on the value of s. In comparison, the Hessian of $G(., s)$ under (4) reads

$$H := 2A^T A + \beta D^T D, \tag{6}$$

hence it is independent of s. This is an important advantage which pushes us to focus on (4).

3. PROPOSED ALGORITHM

In our context, G is convex function with respect to (x, s). Therefore, with an initial guess $(x^{(0)}, s^{(0)})$ for (x, s), we can minimize $G(x, s)$ by first solving

$$G(x^{(0)}, s^{(1)}) \equiv \min_s G(x^{(0)}, \cdot)$$

and then

$$G(x^{(1)}, s^{(1)}) \equiv \min_x G(\cdot, s^{(1)}).$$

We develop an alternating minimization (AM) algorithm in which the function value $G(x^{(n)}, s^{(n)})$ always decreases as n increases. More precisely, the algorithm is stated as follows:

Assume we have $x^{(k)}$ and $s^{(k)}$:

- Solve for $s^{(k+1)} = [s_1^{(k+1)}, s_2^{(k+1)}, \cdots s_r^{(k+1)}]^T$ where $s_i^{(k+1)}$ can be determined by solving

$$\min_{s_i^{(k+1)}} \psi(s_i^{(k+1)}) + (d_i^T x^{(k)} - s_i^{(k+1)})^2. \tag{7}$$

This minimization is very easy since it is performed for each i separately. Notice that the next step does not involve the minimum value $\psi(s_i^{(k+1)})$ but only its argument $s_i^{(k+1)}$.

- Solve for $x^{(k+1)}$

$$Hx^{(k+1)} = 2A^T y + \sum_{i=1}^r s_i^{(k+1)} d_i. \tag{8}$$

We remark that the matrix in the left side above is H given in (6) and it is constant, whereas the right-side is easily updated at each iteration. The costly stage in the algorithm is the inversion of H. To this end, we propose to use pertinent preconditioning of H in order to speed up the computation of $x^{(k+1)}$.

4. PRECONDITIONING OF G

Usually, image reconstruction problems are to recover x from y and the corresponding system matrix A. In these applications, these system matrices are Toeplitz-like, [13]. In this paper, we will consider the preconditioned conjugate gradient method to solve the linear system in (8) with Toeplitz-like coefficient matrix. Optimal transformed based matrices are used to precondition Toeplitz-like matrices in conjugate gradient iterations [9]. Part of their motivation was to exploit the fast inversion of transform based matrices. Numerical results suggest that the method converges very fast for a wide range of Toeplitz-like observation operators A.

4.1. Circulant Preconditioners

For a p-by-p matrix B, the optimal circulant preconditioner $c_F(B)$ is defined to be the minimizer of $\|B - C\|_F$ over all p-by-p circulant matrices C; see for instance [9]. Here $\|\cdot\|_F$ denotes the Frobenius norm. In each iteration, we solve the preconditioned system:

$$
\begin{aligned}
& (2c_F(A)^T c_F(A) + \beta c_F(D^T D))^{-1} Hx^{(k+1)} \\
= \; & (2c_F(A)^T c_F(A) + \beta c_F(D^T D))^{-1} \cdot \\
& \cdot [2A^T y + \sum_{i=1}^r s_i^{(k+1)} d_i].
\end{aligned}
$$

The main cost per iteration is the two matrix-vector multiplications: $(2c_F(A)^T c_F(A) + \beta c_F(D^T D))^{-1} v$ and $(2A^T A + \beta D^T D)w$. Since circulant matrices can be diagonalized by discrete Fourier matrices F, and hence the inversion of p-by-p circulant systems can be done in $O(p \log p)$ operations. The cost of computing $(2A^T A + \beta D^T D)w$ can be computed by embedding A into a larger circulant matrix and then carrying out the multiplication. The multiplication thus requires $O(p \log p)$ operations [9].

4.2. Cosine Transform Based Preconditioners

Besides circulant preconditioners for Toeplitz-like blurring matrices, we note that discrete cosine transform matrices can diagonalized blurring matrices from symmetric blurring functions. For nonsymmetric blurring functions, we define the optimal cosine transform preconditioners $c_C(B)$ to be the minimizer of $\|B - Q\|_F$ over all Q that can be diagonalized by cosine transform matrices. The preconditioners can speed up the convergence of iterative methods.

In view of the results in [13], it is easy to find the optimal cosine transform preconditioners for blurring matrices generated by nonsymmetric blurring functions. We just take the symmetric part of the blurring functions and form the (block) Toeplitz-plus-Hankel matrices. It has been shown theoretically and numerically that if the blurring function

is close to symmetric, then the cosine transform precondi-
tioner $c_C(A)$ will be a good approximation (hence a good
preconditioner) to A.

5. APPLICATIONS: IMAGE RECONSTRUCTION

In the experiments, we consider reconstructing high resolu-
tion images from multiple under-sampled, shifted, degraded
frames with sub-pixel displacement errors [3]. The result-
ing blurring matrices are spatially variant [3]. The classical
Tikhonov regularization and the Neumann boundary condi-
tion are used in the reconstruction process. The precondi-
tioners are derived by taking the cosine transform approx-
imation of the blurring matrices. Here we briefly describe
the model in high resolution image reconstruction. Details
can be found in [3]. Consider a sensor array with $L_1 \times L_2$
sensors in which each sensor has $N_1 \times N_2$ sensing elements
(pixels) and the size of each sensing element is $T_1 \times T_1$.
Our aim is to reconstruct an image of resolution $M_1 \times M_2$,
where $M_1 = L_1 \times N_1$ and $M_2 = L_2 \times N_2$. In order to have
enough information to resolve the high-resolution image,
there are sub-pixel displacements between sensors. In the
ideal case, the sensors are shifted from each other by a value
proportional to $T_1/L \times T_2/L$. However, in practice there
can be small perturbations around these ideal sub-pixel lo-
cations due to imperfections of the mechanical imaging sys-
tem. Thus, for $l_1, l_2 = 0, 1, \cdots, L-1$ with $(l_1, l_2) \neq (0,0)$,
the horizontal and vertical displacements $d^x_{l_1 l_2}$ and $d^y_{l_1 l_2}$ of
the $[l_1, l_2]$-th sensor with respect to the $[0,0]$-th reference
sensor are given by

$$d^x_{l_1 l_2} = \frac{T_1}{L}(l_1 + \epsilon^x_{l_1 l_2}) \quad \text{and} \quad d^y_{l_1 l_2} = \frac{T_2}{L}(l_2 + \epsilon^y_{l_1 l_2}).$$

Here $\epsilon^x_{l_1 l_2}$ and $\epsilon^y_{l_1 l_2}$ denote respectively the normalized hori-
zontal and vertical displacement errors. Let f be the original
scene. Then the observed low-resolution image $g_{l_1 l_2}$ from
the (l_1, l_2)-th sensor is modeled by:

$$g_{l_1 l_2}[n_1, n_2]$$
$$= \int_{T_2(n_2-\frac{1}{2})+d^y_{l_1 l_2}}^{T_2(n_2+\frac{1}{2})+d^y_{l_1 l_2}} \int_{T_1(n_1-\frac{1}{2})+d^x_{l_1 l_2}}^{T_1(n_1+\frac{1}{2})+d^x_{l_1 l_2}} f(x_1, x_2) dx_1 dx_2$$
$$+ \eta_{l_1 l_2}[n_1, n_2].$$

for $n_1 = 1, \ldots, N_1$ and $n_2 = 1, \ldots, N_2$. Here $d^x_{l_1 l_2}$ and
$d^y_{l_1 l_2}$ of the $[l_1, l_2]$-th sensor with respect to the $[0,0]$-th ref-
erence sensor are the horizontal and vertical displacements,
and $\eta_{l_1 l_2}$ is the noise corresponding to the (l_1, l_2)-th sensor.
We interlace the low-resolution images to form an $M_1 \times M_2$
image by assigning

$$g[L(n_1 - 1) + l_1, L(n_2 - 1) + l_2] = g_{l_1 l_2}[n_1, n_2].$$

Here g is an $M_1 \times M_2$ image and is called the *observed high-
resolution image*. The blurring matrix corresponding to the

β	α	Iter.
1e-2	(5,10,20,40)	(20,14,9,5)
1e-3	(5,10,20,40)	(20,15,9,5)
1e-4	(5,10,20,40)	(28,16,9,5)

Table 1. Number of AM iterations for SNR=40dB.

β	α	Iter.
1e-2	(5,10,20,40)	(20,15,9,5)
1e-3	(5,10,20,40)	(17,13,9,5)
1e-4	(5,10,20,40)	(15,11,8,5)

Table 2. Number of AM iterations for SNR=50dB.

(l_1, l_2)-th sensor under the Neumann boundary condition is
given by

$$H_{l_1 l_2}(\epsilon_{l_1, l_2}) = H^x_{l_1 l_2}(\epsilon^x_{l_1, l_2}) \otimes H^y_{l_1 l_2}(\epsilon^y_{l_1, l_2}).$$

The blurring matrix for the whole sensor array is made up of
blurring matrices from each sensor. The resulting blurring
matrices are spatially variant. However, we remark the op-
timal cosine transform based preconditioners are very good
approximation for these blurring matrices.

6. NUMERICAL RESULTS

In the tests, we concentrate on Huber regularizations which
are defined using

$$\phi(t) = \begin{cases} \frac{1}{2}t^2 & \text{if} \quad |t| \leq \alpha \\ \alpha|t| - \frac{1}{2}\alpha^2 & \text{if} \quad |t| > \alpha \end{cases} \quad (9)$$

The threshold parameter α controls the size of the discon-
tinuities modeled by the prior by providing a less severe
edge penalty. The stopping criteria of the alternating min-
imization (AM) method and the preconditioned conjugate
gradient (PCG) method are $||x^{(k)} - x^{(k-1)}||_2 < 20$ and
$||r^{(j)}||/||r^{(0)}|| < 10^{-6}$ respectively, where $r^{(j)}$ is the nor-
mal equations residual after j iterations.

Using the image reconstruction application in Section
5, we first show the efficiency of our method. The number
of AM iterations are listed in Tables 1 and 2. When α in-
creases, the number of AM iterations decreases. However,
when β changes, the numbers of AM iterations are almost
the same for different α.

We also note that the numbers of iterations of using the
PCG method for inverting the matrix in (6) are almost the
same for each AM iteration. In Tables 3 and 4, the total
numbers of CG and PCG iterations required to solve the
linear systems in each AM step are listed. Here we used
the optimal cosine transform based preconditioners for this
image reconstruction problem. We see from the tables that
it takes significantly more iterations using the CG method
that those using the PCG method.

β	CG	PCG
1e-2	(480,336,216,120)	(120,84,54,30)
1e-3	(1180,885,531,295)	(140,105,63,35)
1e-4	(4172,2384,1341,745)	(252,144,81,45)

Table 3. Number of CG iterations for solving the system with the coefficient matrix in (6) when SNR=40dB for $\alpha = (5, 10, 20, 40)$.

β	CG	PCG
1e-2	(480,360,216,120)	(120,84,54,30)
1e-3	(969,741,513,285)	(140,105,63,35)
1e-4	(1995,1463,1064,665)	(252,144,81,45)

Table 4. Number of CG iterations for solving the system with the coefficient matrix in (6) when SNR=50dB for $\alpha = (5, 10, 20, 40)$.

Next we compare the number of iterations required to solve the linear systems with the coefficient matrices in (5) and (6). We find that the number of iterations of using CG or PCG for solving the linear system with the coefficient matrix in (5) is in average more than 300. This number is much greater than those required for solving linear system with the coefficient matrix in (6). This demonstrates the effectiveness of our new formulation.
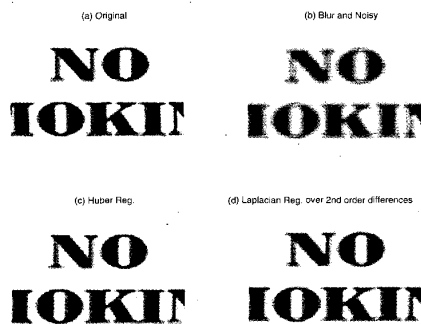

Fig. 1. Results of the first test image.

Finally, in Figure 1, the part of the original image (a), its observed high-resolution (b) and reconstructed images using Huber (c) and Laplacian (d) regularizations are shown in Figure 1. It is clear that the image using the Huber regularization is much better reconstructed than that using the Laplacian regularization.

7. REFERENCES

[1] G. Aubert and L. Vese, "A variational method in image recovery," *SIAM Journal on Numerical Analysis*, vol. 34, no. 5, pp. 1948–1979, 1997.

[2] Michael Black and Anand Rangarajan, "On the unification of line processes, outlier rejection, and robust statistics with applications to early vision," *International Journal of Computer Vision*, vol. 19, no. 1, pp. 57–91, 1996.

[3] N. Bose and K. Boo, "High-resolution Image Reconstruction with Multisensors," International Journal of Imaging Systems and Technology, vol 9, pp. 294–304, 1998.

[4] C. Bouman and K. Sauer, "A generalized Gaussian image model for edge-preserving MAP estimation," *IEEE Transactions on Image Processing*, vol. IP-2, no. 3, pp. 296–310, July 1993.

[5] Ch. Bouman and K. Sauer, "A unified approach to statistical tomography using coordinate descent optimization," *IEEE Transactions on Image Processing*, vol. 5, no. 3, pp. 480–492, Mar. 1996.

[6] P. Charbonnier, Laure Blanc-Féraud, Gilles Aubert, and Michel Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 298–311, Feb. 1997.

[7] L. D. Cohen, "Auxiliary variables and 2-step iterative algorithms in computer vision problems," *Journal of Mathematical Imaging and Vision*, vol. 6, pp. 59–83, 1996.

[8] A. H. Delanay and Y Bressler, "Globally convergent edge-preserving regularized reconstruction: an application to limited-angle tomography," *IEEE Transactions on Image Processing*, vol. 7, pp. 204–221, Feb. 1998.

[9] R. Chan and M. Ng, "Conjugate gradient methods for Toeplitz systems," *SIAM Review*, vol 38, pp. 427–482, 1996.

[10] Donald Geman and Georges Reynolds, "Constrained restoration and recovery of discontinuities," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-14, no. 3, pp. 367–383, Mar. 1992.

[11] Donald Geman and Chengda Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Transactions on Image Processing*, vol. IP-4, no. 7, pp. 932–946, July 1995.

[12] P. Kornprobst, R. Deriche, and G. Aubert, "Image sequence analysis via partial differential equations," *Journal of Mathematical Imaging and Vision*, vol. 11, no. 1, pp. 5–26, Oct. 1999.

[13] M. Ng, R. Chan and W. Tang, "A fast algorithm for deblurring models with Neumann boundary conditions," *SIAM J. Sci. Comput.*, vol 21 pp. 851–866, 1999.