Practice 4
Non smooth optimization and application to image processing

Jean-François AUJOL & Nicolas Papadakis

IMB, Université Bordeaux
351 Cours de la libération, 33405 Talence Cedex, FRANCE

Email : jean-francois.aujol@math.u-bordeaux.fr

1 Convex optimization

1.1 Proximal operators and other definitions

We remind some mathematical concepts coming from convex analysis. X and Y are two finite-dimensional real vector spaces embedded with an inner product $\langle .,. \rangle$ and the associated norm $\|.\| = \sqrt{\langle .,. \rangle}$. We introduce a continuous linear operator $K : X \to Y$ with respect to the induced norm

$$\|K\| = \max \{\|Kx\| \mid x \in X \text{ and } \|x\| \leq 1\} . \quad (1)$$

Let $F : X \to [0, +\infty)$ and $G : X \to [0, +\infty)$ be two proper, convex, lower semi-continuous (l.s.c.) functions.

We define the Legendre-Fenchel conjugate of F by $F^*(y) = \max_{x \in X} (\langle x, y \rangle - F(x))$.

We recall that the subdifferential of F, denoted by ∂F, is defined by

$$\partial F(x) = \{p \in X \text{ such that } F(y) \geq F(x) + \langle p, y - x \rangle \ \forall y\} . \quad (2)$$

The proximity operator is defined by:

$$y = (I + h\partial F)^{-1}(x) = \text{prox}_h^F(x) = \arg\min_u \left\{ \frac{\|u - x\|^2}{2h} + F(u) \right\} \quad (3)$$

Notice computing the proximal operator is itself a minimization problem. As we will see later, the proximal operator is sometimes straightforward to compute but may in other cases require the use of an iterative algorithm.

We can define the Legendre Fenchel transform of F:

$$F^*(v) = \sup_u \left(\langle u, v \rangle - F(u) \right) \tag{4}$$

The Moreau identity reads:

$$x = \mathrm{prox}_h^F(x) + h\mathrm{prox}_{1/h}^{F^*}(x/h) \tag{5}$$

1.2 Primal algorithms: forward-backward splitting and projected gradient

1.2.1 Forward-backward splitting and an accelerated version (FISTA)

The Forward-Backward algorithm was designed to solve the unconstrained minimization problem:

$$\min_{x \in X} F(x) + G(x) \tag{6}$$

where F is a convex $C^{1,1}$ function, with ∇F L-Lipschitz, and G a simple convex l.s.c. function (simple means that the *proximity* operator of G is easy to compute).

The Forward-Backward algorithm reads:

Algorithm 1 Forward-Backward algorithm

- *Initialization:* choose $x_0 \in X$.
- *Iterations ($k \geq 0$):* update x_k as follows:

$$x_{k+1} = (I + h\partial G)^{-1}(x_k - h\nabla F(x_k)). \tag{7}$$

This algorithm is known to converge provided $h \leq 1/L$. In terms of objective functions, the convergence rate is of order $1/k$.

It has been shown by Nesterov, and by Beck and Teboulle, that it could be modified so that a convergence rate of order $1/k^2$ is obtained. The following algorithm, proposed by Beck and Teboulle and called FISTA, converges provided $h \leq 1/L$:

Algorithm 2 FISTA

- *Initialization:* choose $x_0 \in X$; set $y_1 = x_0,\ t_1 = 1$.
- *Iterations ($k \geq 1$):* update $x_k,\ t_{k+1},\ y_{k+1}$ as follows:

$$\begin{cases} x_k = & (I + h\partial G)^{-1}(y_k - h\nabla F(y_k)) \\ t_{k+1} = & \frac{1+\sqrt{1+4t_k^2}}{2} \\ y_{k+1} = & x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1}) \end{cases} \tag{8}$$

Notice that both algorithms require the computation of the proximal operator $(I + h\partial G)^{-1}$, which may itself be computed via a Forward-Backward algorithm or FISTA.

1.2.2 Projected gradient algorithm

A particular case of Problem (6) is the constrained minimization of a differentiable function F, i.e. when G is the indicator function of a closed convex non-empty set C:

$$G(x) = \chi_C(x) = \begin{cases} 0 & \text{if } x \in C, \\ +\infty & \text{otherwise.} \end{cases} \tag{9}$$

The projected gradient algorithm is exactly Algorithm 1 replacing G by χ_C. Indeed, one easily checks that $(I + h\partial G)^{-1} = P_C$, the orthogonal projection onto C. In this case, Algorithm (2) is an accelerated projected gradient method that converges provided $h \leq 1/L$.

Algorithm 3 Accelerated Projection algorithm

- *Initialization:* choose $x_0 \in X$; set $y_1 = x_0,\ t_1 = 1$.
- *Iterations ($k \geq 1$):* update $x_k,\ t_{k+1},\ y_{k+1}$ as follows:

$$\begin{cases} x_k = & P_C(y_k - h\nabla F(y_k)) \\ t_{k+1} = & \frac{1+\sqrt{1+4t_k^2}}{2} \\ y_{k+1} = & x_k + \frac{t_k - 1}{t_{k+1}}(x_k - x_{k-1}) \end{cases} \tag{10}$$

1.3 A primal-dual algorithm: Chambolle Pock Algorithm

The saddle-point problem

$$\min_{x \in X} \max_{y \in Y} \left(\langle Kx, y \rangle + G(x) - F^*(y) \right) \tag{11}$$

is a primal-dual formulation of the nonlinear primal problem $\min_{x \in X} \left(F(Kx) + G(x) \right)$ or its dual $\max_{y \in Y} \left(-G^*(-K^* y) + F^*(y) \right)$.

Algorithm 4 Chambolle-Pock algorithm

- *Initialization:* choose $\tau, \sigma > 0$, $(x_0, y_0) \in X \times Y$, and set $\bar{x}_0 = x_0$.
- *Iterations ($k \geq 0$):* update $x_k,\ y_k,\ \bar{x}_k$ as follows:

$$\begin{cases} y_{k+1} = (I + \sigma \partial F^*)^{-1}(y_k + \sigma K \bar{x}_k) \\ x_{k+1} = (I + \tau \partial G)^{-1}(x_k - \tau K^* y_{k+1}) \\ \bar{x}_{k+1} = 2x_{k+1} - x_k. \end{cases} \tag{12}$$

The convergence of this algorithm is proved:

Theorem 1 *Assume Problem* (11) *has a saddle point. Choose τ and σ such that $\tau\sigma\|K\|^2 < 1$, and let $(x_n, \bar{x}_n, y_n)$ be defined by (12). Then there exists a saddle point (x^*, y^*) such that $x^n \to x^*$ and $y^n \to y^*$.*

Notice that Algorithm (11) can be used even when both F and G are not smooth.

2 Applications

2.1 Computation of some proximal operators

In the following, you will need to be able to use the proximal operator of some convex functions. We recall that the proximal operator is defined by:

$$y = (I + h\partial F)^{-1}(x) = \mathrm{prox}_h^F(x) = \arg\min_u \left\{ \frac{\|u - x\|^2}{2h} + F(u) \right\}$$

Compute the following proximal operators:

1.
$$F(u) = \frac{1}{2}\|u\|_2^2$$
Show that
$$\text{prox}_h^F(x) = \frac{x}{1+h}$$

2.
$$F(u) = \frac{1}{2}\|u - f\|_2^2$$
Show that
$$\text{prox}_h^F(x) = \frac{x + hf}{1+h}$$

3.
$$F(u) = \frac{1}{2}\|Ku - f\|_2^2$$
Show that
$$\text{prox}_h^F(x) = (Id + hK^*K)^{-1}(x + hK^*f)$$

4.
$$F(u) = \|u\|_1$$
Show that
$$\text{prox}_h^F(u) = ST(u, h)$$
with $ST(u,h)$ the Soft-Thresholding of u with parameter h, i.e. $ST(u,h) = u - h$ if $u > h$, $ST(u,h) = u + h$ if $u < -h$, and $ST(u,h) = 0$ if $|u| \leq h$.

5.
$$F(u) = \|\nabla u\|_1$$
Show that $y = \text{prox}_h^F(x)$ if and only if $y = x - h\,div(z)$ with z solution of
$$\min_{\|z\|_\infty \leq 1} \|div(z) + x/h\|_2^2$$
This last problem can easily be solved by a projection algorithm.
Hint: $H(z) = \|div(z) + x/h\|_2^2$. Then $\nabla H(z) = -2\nabla(div(z) + x/h)$. And if $C = \{z\ ,\ \|z\|_\infty \leq 1\}$, then the proximal operator of χ_C is the orthogonal projection onto C.

6. Write matlab functions to compute proximal operators 1, 2, 4, and 5.

2.2 Image restoration (gaussian noise)

We use the model:

$$\inf_u \lambda \|f - u\|_2^2 + \|\nabla u\|_1 \tag{13}$$

Using the Forward-Backward algorithm (FB), write an algorithm computing the solution of this problem.

Run the algorithm on different type of images, with different level of noise (zero mean gaussian noise with standard deviation σ).

Comments ?

Try the convergence sped-up (FISTA). Plot the value of the functional with respect to the number of iterations. Comments ?

2.3 Image restoration (salt and peper noise)

We use the model:

$$\inf_u \lambda \|f - u\|_1^1 + \|\nabla u\|_1 \tag{14}$$

Using the primal dual algorithm by Chambolle-Pock, write an algorithm to compute the solution of the above problem.

Salt and peper noise means that some pixels of the image are arbitrary set to 0 or 255.

Run the algorithm on different type of images, with different level of noise.

Comments ?

Compare with the $TV - L2$ model.

2.4 Image deconvolution (gaussian noise)

We use the model:

$$\inf_u \lambda \|f - Ku\|_2^2 + \|\nabla u\|_1 \tag{15}$$

where K is a convolution operator (K will be a gaussian convolution operator).

Using the Forward-Backward algorithm (FB), write an algorithm computing the solution of this problem.

Run the algorithm on different type of images, with different level of noise (zero mean gaussian noise with standard deviation σ).

Comments ?

Try the convergence sped-up (FISTA). Plot the value of the functional with respect to the number of iterations. Comments ?

2.5 Image deconvolution (salt and peper noise)

We use the model:

$$\inf_u \lambda \|f - Ku\|_1 + \|\nabla u\|_1 \tag{16}$$

where K is a convolution operator (K will be a gaussian convolution operator).

Using the primal dual algorithm by Chambolle-Pock, write an algorithm to compute the solution of the above problem (hint: you may use the FFT to compute proximal operator 3).

Run the algorithm on different type of images, with different level of noise.

Comments ?

Compare with the previous model.

2.6 Image inpainting

We want to compare the models:

$$\inf_u \lambda \|f - Ku\|_2^2 + \|\nabla u\|_1 \tag{17}$$

$$\inf_u \lambda \|f - Ku\|_1 + \|\nabla u\|_1 \tag{18}$$

where K is a masking operator.

Compare the two methods on the images of practice 2. Comments ?