Code Documentation

Lénaic Chizat

June 23, 2016

Abstract

Some technical details about the implementation of the Optimal Transport with Julia toolbox.

1 Projection on the set of constraints

1.1 No source term

In the case where there is no source term, the set of constraints reads

$$\mathcal{C} = \left\{ x := (\rho, m)^T \in \mathbb{R}^N \mid Ax = b_0, A = \left[\frac{\mathbf{div}}{b}\right] \text{ and } b_0 = \left[\frac{0}{b_c}\right] \right\} \tag{1}$$

where b_c is a column vector of size r (the number of constraints). The proximal operator of the function $\iota_{\mathcal{C}}$ is the projector on $\mathcal{C}$, denoted $\mathcal{P}_{\mathcal{C}}$. It can explicitly computed as

$$\mathcal{P}_{\mathcal{C}}(x) = x + A^*(AA^*)^{-1}(b_0 - Ax) \tag{2}$$

where A^*, the conjugate operator of A, reads $A^* = [-\mathbf{grad}, b^*]$. As the linear operator $b : \mathbb{R}^N \to \mathbb{R}^r$ selects the boundaries of x, we have $bb^* = Id_r$, the identity operator in $\mathbb{R}^r$ and $b^*b := I_b$, an linear endomorphism in $\mathbb{R}^N$ which leave the components of x on the boundaries unchanged and sets to zero the remaining coefficients. It follows readily

$$AA^* = \left[\begin{array}{c|c} -\Delta & \mathbf{div}\, b^* \\ \hline -b\,\mathbf{grad} & Id_r \end{array}\right] \left(= \left[\begin{array}{c|c} a & b \\ \hline c & d \end{array}\right]\right) \tag{3}$$

where we introduced the Laplacian operator $\Delta = \mathbf{div}\,\mathbf{grad}$.

We introduce S the Schur complement of AA^*, namely $S = -\Delta - \mathbf{div} b^* Id_r^{-1}(-b\mathbf{grad})$ — we recall that it is generally defined as $S = a - bd^{-1}c$, using the bloc matrix notation above. It reduces to $S = -\mathbf{div}(Id_N - I_b)\mathbf{grad}$. Finding u such that $p = Su$ is possible, a particular solution is to solve a Poisson equation with Neumann boundary conditions (the gradient should vanish at the boundaries so that the I_b operator can be neglected). We denote $u = S^{-1}p$ this solution. With the Schur complement, the inverse of a 4×4 invertible bloc matrix reads

$$\left[\begin{array}{c|c} a & b \\ \hline c & d \end{array}\right]^{-1} = \left[\begin{array}{c|c} I & 0 \\ \hline -d^{-1}c & I \end{array}\right] \left[\begin{array}{c|c} S^{-1} & 0 \\ \hline 0 & d^{-1} \end{array}\right] \left[\begin{array}{c|c} I & -bd^{-1} \\ \hline 0 & I \end{array}\right] \tag{4}$$

Remark 1. *If we make sure that x satisfies the boundary conditions, the projector has a simpler expression. In the algorithm, we just need to make sure that the first input x_0 satisfies the BC-conditions and that all our operators preserve this property. In that case, the projector can be written in a simple form : for $x \in \mathbb{R}^N$ such that*

$bx = b_c$ and $w := -\mathbf{div}(x)$, we have

$$\begin{aligned}
\mathcal{P}^r_{\mathcal{C}}(x) &= x + A^*(AA^*)^{-1}(b_0 - Ax) && (5)\\
&= x + A^*\left[\frac{S^{-1}w}{-Id_r^{-1}(-b\,\mathbf{grad})S^{-1}w}\right] && (6)\\
&= x + \left\{-\mathbf{grad}S^{-1} + I_b\mathbf{grad}S^{-1}\right\}(w) && (7)\\
&= x - (Id_N - I_b)\mathbf{grad}S^{-1}(-\mathbf{div}(x)) && (8)
\end{aligned}$$

when the subscript r indicates that this projector is only valid on a certain restricted subspace of $\mathbb{R}^N$

1.2 With source term

With a source term, the line of reasoning is similar. We use the same notations as before and the set of constraint now reads

$$\mathcal{C} = \left\{ x := (\rho, m, \zeta)^T \in \mathbb{R}^N \mid Ax = b_0, A = \left[\frac{\mathbf{div} - f}{b}\right] \text{ and } b_0 = \left[\frac{0}{b_c}\right] \right\} \quad (9)$$

where we introduced f, the linear operator $\mathbb{R}^N \to \mathbb{R}^M$ which selects the coefficients in x describing ζ. Also now, **div** has a slightly different meaning: it only acts on the components (ρ, m) of x.

Now A^* the conjugate operator of A reads $A^* = [-\mathbf{grad} - f^*, b^*]$—again, **grad** is a "restricted" gradient. Notice that f behaves similarly to b since $ff^* = Id_M$ and $f^*f := I_\zeta$, an operator which sets all components of x to zero except those describing ζ. Thus,

$$AA^* = \left[\begin{array}{c|c} P & (\mathbf{div} - f)b^* \\ \hline -b\,(\mathbf{grad} + f^*) & Id_r \end{array}\right] \quad (10)$$

where we define $P := (\mathbf{div} - f)(-\mathbf{grad} - f^*) = ff^* - \Delta$.

Here the Schur complement of AA^* is $S = -(\mathbf{div} - f)(Id_M - I_b)(\mathbf{grad} + f^*)$ which reduces to $S = -\mathbf{div}(Id_N - I_b)\mathbf{grad} + Id_M$, by noticing that some products of operators vanish because there is no boundary constraint on ζ. Finding u such that $Su = p$ is possible, a particular solution is to solve a the equation $-\Delta u + u = p$ with Neumann boundary conditions, which can still be efficiently performed in the Fourier domain. We denote $u = S^{-1}p$ this solution.

We still make the assumption that x already satisfies the boundary constraints. Thus, for $x \in \mathbb{R}^N$ such that $bx = b_c$ and $w := -\mathbf{div}(x) + f(x)$, we have

$$\begin{aligned}
\mathcal{P}^r_{\mathcal{C}}(x) &= x + A^*(AA^*)^{-1}(b_0 - Ax) && (11)\\
&= x + A^*\left[\frac{S^{-1}w}{b(\mathbf{grad} + f^*)S^{-1}w}\right] && (12)\\
&= x + \left\{(-\mathbf{grad} - f^*)S^{-1} + b^*b(\mathbf{grad} + f^*)S^{-1}\right\}(w) && (13)\\
&= x - (I_M - I_b)(\mathbf{grad} + f^*)S^{-1}(f - \mathbf{div})(x) && (14)
\end{aligned}$$

1.3 No assumption on the input

If we relax the assumption that the input satisfies the boundary conditions, the generalized projector $\mathcal{P}_{\mathcal{C}}$ can still be written in a simple form using the previous expression

$$\begin{aligned}
\mathcal{P}_{\mathcal{C}}(x) &= \mathcal{P}_{\mathcal{C}}(x + b^*(b_c - bx)) && (15)\\
&= \mathcal{P}^r_{\mathcal{C}} \circ \mathcal{P}_B(x) && (16)
\end{aligned}$$

where $\mathcal{P}_B$ is the projection on the boundary contraints. This can be checked by using the the factorized form of the inverse AA^* matrix (see above).

2 Solve Discrete Poisson equation in the Fourier domain

In this section, we deal with 2D discrete fields, but the results straightforwardly generalize to fields with an arbitrary number of discrete variables. p and u are discrete functions i.e $p, u \in R^{N\times N}$, with sampling rate $1/h$ on both variables, and Δ refers to the discrete Laplacian operator, defined by

$$\Delta(u)[i,j] = \frac{1}{h^2}\left(u[i+1,j] + u[i-1,j] + u[i,j+1] + u[i,j-1] - 4u[i,j]\right)$$

(the values on boundary are dealt with in the following). In the following, the index of the centered grid go from 1 to N and those of the staggered grid from $1/2$ to $N+1/2$.

2.1 With no source term

The problem $p = Su$ with unknown u recasts into finding u such that $\Delta(u) + p = 0$. We can solve this equation in the Fourier domain as

$$\left(4 - 2\cos(2i\pi m/N) - 2\cos(2i\pi n/N)\right)\hat{u}[m,n] = h^2\hat{p}[m,n]\,. \tag{17}$$

Remark that the value of $\hat{u}[1,1]$ is not determined and needs to be arbitrarily defined.

Recall that we are looking for a solution with vanishing gradient on the boundaries. The DCT-II implies that the extension u_s of u is even around $i,j = 1/2$ and even around $i,j = N+1/2$. Thus, the gradient of u_s restricted to the staggered grid $(i,j) \in \{1/2,\dots,N+1/2\}^2$ vanishes on the boundaries of the staggered grid since u_s admits local extrema there. Finally, transform back in the time domain with the DCT-III, which inverts the DCT-II transform.

As a remainder, the DCT-II transform of u is defined as

$$\hat{u}[k] = \sum_{n=1}^{N} u[n]\cos\left[\frac{\pi}{N}(n+\frac{1}{2})k\right] \tag{18}$$

2.2 With a source term

The problem $p = Su$ recasts into finding u such that $\Delta(u) - u + p = 0$. In the Fourier domain, this equation reads

$$\left(5 - 2\cos(2i\pi m/N) - 2\cos(2i\pi n/N)\right)\hat{u}[m,n] = h^2\hat{p}[m,n]\,. \tag{19}$$

The vanishing gradient BC on the staggered grid remains and we still use the DCT-II/DCT-III transforms. The slight difference to the previous case is that now the term $\hat{u}[1,1]$ is uniquely determined by the equation.

3 Projection on the interpolation constraints

We won't divide this section between source term / no source term, because the interpolation of the source term in straightforward and independent of the other term: as ζ is always defined on the centered grid, the projection simply consists in taking the mean.

The interpolation constraint set is

$$\mathcal{H} = \{(U,V) \mid V = I(U), V \in \mathcal{E}_c, \text{ and } U \in \mathcal{E}_s, \text{plus BC}\}$$

Where I is the interpolation operator. Let us derive the projector on this set

$$\begin{aligned}(U,V) &= \mathscr{P}_{\mathcal{H}}(U_0,V_0) &(20)\\ &= \arg\min_{V=I(U)} \frac{1}{2}|U_0-U|^2 + \frac{1}{2}|V_0-V|^2 &(21)\end{aligned}$$

We obtain

$$U = Q^{-1}(U_0 + I^* V_0)$$

where $Q = I^* I + I_d$. Precompute Q^{-1}; may use LU factorization as Q is tridiagonal. Full matrix multiplication : longest step.

4 Proximal of the functional with Fisher-Rao metric

4.1 General case

The discretized functional reads

$$J_\lambda(\rho, m, \zeta) = \sum_{x,t} \frac{|m(x,t)|^2 + \lambda \zeta^2}{2\rho(x,t)}$$

The proximal is the argmin of a minimization problem that can be solved independently on each point of the centered grid. The optimality conditions yields the following system:

$$\left\{ \begin{array}{rcl} P(\rho) & = & 0 \\ m & = & \tilde{m}\left(1 - \frac{\gamma}{\rho+\gamma}\right) \\ \zeta & = & \tilde{\zeta}\left(1 - \frac{\lambda\gamma}{\rho+\lambda\gamma}\right) \end{array} \right. \tag{22}$$

where P is the fifth-order polynomial

$$P[X] = (X - \tilde{\rho})(X+\gamma)^2(X+\lambda\gamma)^2 - \frac{\gamma}{2}\left[|\tilde{m}|^2(X+\lambda\gamma)^2 + \lambda\tilde{\zeta}^2(X+\gamma)^2\right]$$

where $(\tilde{\rho}, \tilde{m}, \tilde{\zeta})$ is the point where the proximal is evaluated. Its expanded form is

$$\begin{array}{rcl} P[X] & = & X^5 \\ & + & X^4\left[2\gamma(\lambda+1) - \tilde{\rho}\right] \\ & + & X^3\left[\gamma^2(\lambda^2+4\lambda+1) - 2\gamma\tilde{\rho}(1+\lambda)\right] \\ & + & X^2\left[2\lambda\gamma^3(1+\lambda) - \tilde{\rho}\gamma^2(\lambda^2+4\lambda+1) - \frac{\gamma}{2}(|\tilde{m}|^2 + \lambda\tilde{\zeta}^2)\right] \\ & + & X^1\left[\lambda^2\gamma^4 - 2\tilde{\rho}\lambda\gamma^3(\lambda+1) - \lambda\gamma^2(|\tilde{m}|^2 + \tilde{\zeta}^2)\right] \\ & + & X^0\left[-\tilde{\rho}\lambda^2\gamma^4 - \lambda\frac{\gamma^3}{2}(\lambda|\tilde{m}|^2 + \lambda\tilde{\zeta}^2)\right] \end{array}$$

(double checked with SymPy). The proximal is the highest real root of this polynomial. However in practice we always put ourselves in the case of a functional with equal costs using an appropriate rescaling.

4.2 Special case $\lambda = 1$

Using a change of variable, one can always reduce the problem to the case of a functional with equal costs on transport and mass creation ($\lambda = 1$). In that case, the polynomial is

$$P[X] = (X - \tilde{\rho})(X+\gamma)^2 - \frac{\gamma}{2}(|\tilde{m}|^2 + \tilde{\zeta}^2)$$

where $(\tilde{\rho}, \tilde{m}, \tilde{\zeta})$ is the point where the proximal is evaluated. Its expanded form is

$$\begin{array}{rcl} P[X] & = & X^3 \\ & + & X^2\left[2\gamma - \tilde{\rho}\right] \\ & + & X^1\left[\gamma^2 - 2\gamma\tilde{\rho}\right] \\ & + & X^0\left[-\frac{\gamma}{2}(|\tilde{m}|^2 + \tilde{\zeta}^2) - \gamma^2\tilde{\rho}\right] \end{array}$$

Centered grid dimensions	$Prox_J(\lambda = 1)$	$Proj_C$	$Proj_I$ Q inverted	$Proj_I$ LU decomp.
$(16, 16, 8)$	3,6 ms	6,1 ms	11,2 ms	11,0 ms
$(120, 120, 20)$	0,57 s	0,50 s	0,82 s	0,66 s
$(256, 256, 32)$	5,7 s	4,2 s	8,9 s	4,3 s
$(1024, 32)$	59 ms	30 ms	138 ms	33 ms

Figure 1: CPU times for the core functions of the algorithm.

5 Proximal of the Piccoli-Rossi functional

The discretized functional reads

$$J(\rho, m, \zeta) = \frac{1}{2}\left(\sum_{x,t} |\zeta(x,t)|\right)^2 + \sum_{x,t} \frac{|m(x,t)|^2}{2\rho(x,t)} \tag{23}$$

The proximal can be taken independently with respect to (ρ, m) and to ζ. Let us thus consider the proximal of the function of ζ which is the squared ℓ_1 norm. The sub differential of the squared ℓ_1 norm at x is $sign(x)\|x\|_1$ where $sign(\cdot)$ is the sub differential of the ℓ_1 norm. The proximal x of $\tilde{x}$ taken at $\gamma(|\cdot|)^2$ should satisfy the following optimality condition:

$$x - x^* \in \gamma sign(x^*)\|x^*\|_1 \,.$$

This can be solved for $x^* = soft_\lambda(x)$ where $soft_\lambda(x) = sign(x)\,(\|x\| - \lambda)_+$. The problem recasts in finding a threshold λ such that

$$f_\gamma(\lambda) := \|x - soft_\lambda(x)\|_\infty - \gamma\|soft_\lambda(x)\|_1 = 0\,.$$

Let the indices denote the sorted values of vector $x = (x_{k_1}, \dots, x_{k_N})$, i.e $|x_1| \leq \cdots \leq |x_N|$. Remark that

$$\begin{cases} f(|x_{i+1}|) - f(|x_i|) = (1 + \gamma(N-i))\,(|x_{i+1}| - |x_i|) \\ f(0) = -\gamma\|x\|_1 \\ f(|x_N|) = \|x\|_\infty = |x_N| \\ f'(\lambda) = 1 + \gamma(N-i) \end{cases}$$

So we can compute the proximal with the following algorithm:

1. Sort $(|x_i|)_i$;
2. Compute $f(|x_i|)$ recursively with the formula above (may stop when crossing zero);
3. Find k such that $f(|x_k|) \leq 0 < f(|x_{k+1}|)$;
4. Set $\lambda = |x_k| - \frac{f(|x_k|)}{1+\gamma(N-k)}$;
5. Set $x^* = soft_\lambda(x)$.

In the special case $x = 0$, return $x^* = 0$.