Using Local Regularization for 1D Deblurring Problems

July 2, 2016

1 Preliminaries and Discretization

Let $r \in (0, 1/2)$ be a regularization parameter. For functions $\varphi : [r, 1-r] \times [-r, r] \to \mathbb{R}$ and functions $\eta : [0,1] \to \mathbb{R}$, we define the operators:

$$(A_r\varphi)(x,\rho) = \int_{[-r,r]} k(x+\rho, x+s)\varphi(x,s)\,\mathrm{d}s\,;$$

$$(B_r\eta)(x,\rho) = \int_{[0,1]\setminus[x-r,x+r]} k(x+\rho, s)\eta(s)\,\mathrm{d}s\,;$$

$$(T_r\varphi)(x) = \begin{cases} \varphi(x,0) & x \in [r, 1-r] \\ 0 & \text{otherwise} \end{cases}.$$

Note that $(A_r\varphi), (B_r\eta) : [r, 1-r] \times [-r, r] \to \mathbb{R}$, while $(T_r\varphi) : [0,1] \to \mathbb{R}$. Then define:

$$C_r\varphi = A_r\varphi + B_r T_r\varphi.$$

Suppose:

$$k(t,s) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(t-s)^2/2\sigma^2}.$$

For some $\sigma > 0$.

Let $f^\delta \in L^\infty([0,1])$ be given with $\|f^\delta - f\| < \delta$. Let $\alpha \in L^\infty([0,1])$ have a minimum value $\alpha_{\min}$ which is positive. The local regularization problem $\mathcal{P}^\delta_{r,\alpha}$ is to find a function $\varphi : [r, 1-r] \times [-r, r] \to \mathbb{R}$ with $\|\varphi\|_r^2 < \infty$ which minimizes:

$$\|C_r\varphi - F_r^\delta\|_r^2 + \|\varphi\|_{r,\alpha}^2,$$

where $F_r^\delta(x,\rho) := f(x+\rho)$.

Well now,

$$\|C_r\varphi - F_r^\delta\|_r^2 = \frac{1}{2r}\int_r^{1-r}\int_{-r}^{r} |(C_r\varphi)(x,\rho) - F_r^\delta(x,\rho)|^2\,\mathrm{d}\rho\,\mathrm{d}x$$

$$= \frac{1}{2r} \sum_{q=n+1}^{N-n} \sum_{p=1-n}^{n} \int_{(q-1)/N}^{q/N} \int_{(p-1)/N}^{p/N} |(C_r\varphi)(x,\rho) - f^\delta(x+\rho)|^2 \,\mathrm{d}\rho \,\mathrm{d}x$$

$$= \frac{1}{2r} \sum_{j=1}^{N-2n+1} \sum_{l=1}^{2n} \int_{(j+n-1)/N}^{(j+n)/N} \int_{(l-n-1)/N}^{(l-n)/N} |(C_r\varphi)(x,\rho) - f^\delta(x+\rho)|^2 \,\mathrm{d}\rho \,\mathrm{d}x$$

$$= \frac{1}{2r} \sum_{j=1}^{N-2n+1} \sum_{l=1}^{2n} \frac{1}{N^2} |(C_r\varphi)(\tilde{x}_{j,l},\tilde{\rho}_{j,l}) - f^\delta(\tilde{x}_{j,l}+\tilde{\rho}_{j,l})|^2,$$

For some

$$\tilde{x}_{j,l} \in \left(\frac{j+n-1}{N}, \frac{j+n}{N}\right),$$
$$\tilde{\rho}_{j,l} \in \left(\frac{l-n-1}{N}, \frac{l-n}{N}\right).$$

Define:

$$x_j = \frac{j+n-1}{N}, \quad j \in \{1,\dots,N-2n+1\},$$
$$\rho_l = \frac{l-n-1}{N}, \quad l \in \{1,\dots,2n\}.$$

Now, assuming φ is peicewise constant, i.e.

$$\varphi(x,\rho) = c_{j,l} \quad x \in [x_j, x_{j+1}], \quad \rho \in [\rho_l, \rho_{l+1}],$$

For $j \in \{1,\dots,N-2n\}$ and $l \in \{1,\dots,2n-1\}$. Then

$$\begin{aligned}(A_r\varphi)(x,\rho) &= \int_{-r}^{r} k(x+\rho, x+s)\varphi(x,s)\,\mathrm{d}s \\ &= \sum_{l=1}^{2n} \int_{\rho_l}^{\rho_{l+1}} k(x+\rho,x+s)\varphi(x,s)\,\mathrm{d}s \\ &= \sum_{l=1}^{2n} c_{j,l} \int_{\rho_l}^{\rho_{l+1}} k(x+\rho,x+s)\,\mathrm{d}s\end{aligned}$$

For some $j = j(x) \in \{1,\dots,N-2n\}$. Let:

$$\boxed{I_l(x,\rho) = \int_{\rho_l}^{\rho_{l+1}} k(x+\rho,x+s)\,\mathrm{d}s.}$$

Then

$$(A_r\varphi)(x,\rho) = \sum_{l=1}^{2n} I_l(x,\rho) c_{j(x),l}.$$

Suppose (Case 1) that $x \in [r, 2r]$. Then $x \in [x_{j_1}, x_{j_1+1}]$ for some $j_1 \in \{1, \ldots, n\}$, and $(x + r) \in [x_{j_1+n}, x_{j_1+n+1}]$. In this case,

$$\begin{aligned}
(B_r T_r \varphi)(x, \rho) &= \int_{x+r}^{1-r} k(x + \rho, s)\varphi(s, 0)\,\mathrm{d}s \\
&= \int_{x+r}^{x_{j_1+n+1}} k(x + \rho, s)\varphi(s, 0)\,\mathrm{d}s + \sum_{j=j_1+n+1}^{N-2n} \int_{x_j}^{x_{j+1}} k(x + \rho, s)\varphi(s, 0)\,\mathrm{d}s \\
&= c_{j_1+n,n+1} \int_{x+r}^{x_{j_1+n+1}} k(x + \rho, s)\,\mathrm{d}s + \sum_{j=j_1+n+1}^{N-2n} c_{j,n+1} \int_{x_j}^{x_{j+1}} k(x + \rho, s)\,\mathrm{d}s
\end{aligned}$$

Where we have used $0 = \rho_{n+1} \in [\rho_{n+1}, \rho_{n+2}]$.

Suppose (Case 2) that $x \in (2r, 1 - 2r)$. Then $x \in [x_{j_2}, x_{j_2+1}]$ for some $j_2 \in \{n + 1, \ldots, N - 3n\}$, $(x - r) \in [x_{j_2-n}, x_{j_2-n+1}]$, and $(x + r) \in [x_{j_2+n}, x_{j_2+n+1}]$. In this case,

$$\begin{aligned}
(B_r T_r \varphi)(x, \rho) &= \int_{r}^{x-r} k(x + \rho, s)\varphi(s, 0)\,\mathrm{d}s + \int_{x+r}^{1-r} k(x + \rho, s)\varphi(s, 0)\,\mathrm{d}s \\
&= \sum_{j=1}^{j_2-n-1} c_{j,n+1} \int_{x_j}^{x_{j+1}} k(x + \rho, s)\,\mathrm{d}s + c_{j_2-n,n+1} \int_{x_{j_2-n}}^{x-r} k(x + \rho, s)\,\mathrm{d}s \\
&+ c_{j_2+n,n+1} \int_{x+r}^{x_{j_2+n+1}} k(x + \rho, s)\,\mathrm{d}s + \sum_{j=j_2+n+1}^{N-2n} c_{j,n+1} \int_{x_j}^{x_{j+1}} k(x + \rho, s)\,\mathrm{d}s
\end{aligned}$$

Suppose (Case 3) that $x \in [1 - 2r, 1 - r]$. Then $x \in [x_{j_3}, x_{j_3+1}]$ for some $j_3 \in \{N - 3n + 1, \ldots, N - 2n\}$, and $(x - r) \in [x_{j_3-n}, x_{j_3-n+1}]$. In this case,

$$\begin{aligned}
(B_r T_r \varphi)(x, \rho) &= \int_{r}^{x-r} k(x + \rho, s)\varphi(s, 0)\,\mathrm{d}s \\
&= \sum_{j=1}^{j_3-n-1} c_{j,n+1} \int_{x_j}^{x_{j+1}} k(x + \rho, s)\,\mathrm{d}s + c_{j_3-n,n+1} \int_{x_{j_3-n}}^{x-r} k(x + \rho, s)\,\mathrm{d}s
\end{aligned}$$

In each case, we have:

$$(B_r T_r \varphi)(x, \rho) = \sum_{j=1}^{N-2n} K_j(x, \rho) c_{j,n+1}$$

(Note that, as opposed to the case for A_r, the index j here does not depend on x. The full extent of the x-dependence is in the function $K_j(x, \rho)$.)

Then:

$$
\begin{aligned}
\|C_r\varphi - F_r^\delta\|_r^2 &= \frac{1}{2r}\int_r^{1-r}\int_{-r}^{r} |(C_r\varphi)(x,\rho) - f^\delta(x+\rho)|^2 \,\mathrm{d}\rho\,\mathrm{d}x \\
&= \frac{1}{2r}\int_r^{1-r}\int_{-r}^{r} |(C_r\varphi)(x,\rho) - f^\delta(x+\rho)|^2 \,\mathrm{d}\rho\,\mathrm{d}x \\
&= \frac{1}{2r}\int_r^{1-r}\int_{-r}^{r} \left|\sum_{l=1}^{2n} I_l(x,\rho)c_{j(x),l} + \sum_{j=1}^{N-2n} K_j(x,\rho)c_{j,n+1} - f^\delta(x+\rho)\right|^2 \mathrm{d}\rho\,\mathrm{d}x \\
&\approx \frac{1}{2rN^2}\sum_{i=1}^{N-2n}\sum_{m=1}^{2n}\left|\sum_{l=1}^{2n} I_l(x_i,\rho_m)c_{i,l} + \sum_{j=1}^{N-2n} K_j(x_i,\rho_m)c_{j,n+1} - f^\delta(x_i+\rho_m)\right|^2
\end{aligned}
$$

And:

$$
\begin{aligned}
\|\varphi\|_{r,\alpha}^2 &= \frac{1}{2r}\int_r^{1-r}\alpha(x)\int_{-r}^{r}|\varphi(x,\rho)|^2\,\mathrm{d}\rho\,\mathrm{d}x \\
&= \frac{1}{2rN}\int_r^{1-r}\alpha(x)\sum_{m=1}^{2n}|\varphi(x,\rho_m)|^2\,\mathrm{d}x \\
&\approx \frac{1}{2rN^2}\sum_{i=1}^{N-2n}\sum_{m=1}^{2n}\alpha(x_i)c_{i,m}^2
\end{aligned}
$$

In the case that $x = x_i$, for some $i \in \{1,\ldots,N-2n\}$, in the coefficient $K_j(x,\rho)$ for $(B_r\varphi)(x,\rho)$, the calculation becomes simpler, since the boundary terms previously required are not needed. We can write:

$$
\begin{aligned}
&(B_rT_r\varphi)(x_i,\rho_m) \\
&= \sum_{j=1}^{N-2n}\left\{\begin{array}{cc}
[j\in\{i+n,\ldots,N-2n\}], & i\in\{1,\ldots,n+1\} \\
[j\in\{1,\ldots,i-n-1\}\cup\{i+n,\ldots,N-2n\}], & i\in\{n+2,\ldots,N-3n\} \\
[j\in\{1,\ldots,i-n-1\}], & i\in\{N-3n+1,\ldots,N-2n+1\}
\end{array}\right\} \\
&\qquad c_{j,n+1}\int_{(j+n-1)/N}^{(j+n)/N} k\left(\frac{i+m-2}{N}, s\right)\mathrm{d}s.
\end{aligned}
$$

For A_r and $i < N-2n+1$, we have:

$$
\begin{aligned}
(A_r\varphi)(x_i,\rho_m) &= \sum_{l=1}^{2n} c_{i,l}\int_{\rho_l}^{\rho_{l+1}} k(x_i+\rho_m, x_i+s)\,\mathrm{d}s \\
&= \sum_{l=1}^{2n} c_{i,l}\int_{(l-i-2n)/N}^{(l-i-2n+1)/N} k\left(\frac{i+m-2}{N}, s\right)\mathrm{d}s
\end{aligned}
$$

Let:

$$
\Delta_{p,a} = \int_{(a-1)/N}^{a/N} k((p-1)/N, s)\,\mathrm{d}s,
$$

Where $p \in \{1, \dots, N+1\}$ and $a \in \{1, \dots, N\}$. Then we have:

$$(A_r\varphi)(x_i, \rho_m) = \sum_{l=1}^{2n} \Delta_{i+m-1, l-i-2n+1} c_{i,l}$$

$$(B_r T_r \varphi)(x_i, \rho_m) = \sum_{j=1}^{N-2n} \{\text{conditions}\} \Delta_{i+m-1, j+n} c_{j,n+1}$$

So then:

$$\|C_r\varphi - F_r^\delta\|_r^2 + \|\varphi\|_{r,\alpha}^2$$

$$\approx \frac{1}{2rN^2} \sum_{i=1}^{N-2n} \sum_{m=1}^{2n} \left| \sum_{l=1}^{2n} \Delta_{i+m-1,l-i-2n+1} c_{i,l} + \sum_{j=1}^{N-2n} \{\text{conditions}\} \Delta_{i+m-1,j+n} c_{j,n+1} - f^\delta(x_i + \rho_m) \right|^2$$

$$+\alpha(x_i)|c_{i,m}|^2$$

$$= \frac{1}{2rN^2} \sum_{i=1}^{N-2n} \mathcal{J}_i(c)$$

We can consider the term $\mathcal{J}_i(c)$, since it is the sum of squares of various things, in terms of the square norm of some vector. The paper uses the same letters for this vector as for the component terms, which could potentially be confusing.

2 Algorithm 2

For each i, the term $\mathcal{J}_i(c)$ depends on (some) of the entries in the the column $c_{\cdot,n+1}$. This column represents values $\varphi(x, 0)$, i.e. the desired data u^δ when unperturbed by the shift parameter ρ. Additionally, $\mathcal{J}_i(c)$ depends on the row $c_{i,\cdot}$, which represents values $\varphi(x_i, \rho)$ for the possible ρ.

The idea behind local regularization algorithm 2, is as follows.

1. Idenfify a permutation of the values $\{1, \dots, N-2n\}$. I.e., a sequence $i_1, \dots, i_{N-2n}$ of values in that set, with no repetitions. Local regularization is a *sequential* algorithm, not an iterative one. It is generally presumed that $i_t = t,\ t \in \{1, \dots, N-2n\}$—the identity permutation—however, it is possible, in my view, that a different sequence could produce slightly different results. As the algorithm is supposed to converge anyways for increasingly small grid size, these differences could only exist if the grid size could not be made very small for some reason.

2. Initialize the matrix c. As the problem we are concerned with is the deblurring problem, the "observed data space" is the same as the "true object space", so we could initialize with the observed data f. In symbols, we could let $c_{i,m} = f^\delta(x_i + \rho_m)$. Alternately, we could let $c_{i,m} = f^\delta(x_i)$. Due to the nature of the algorithm, it is actually somewhat ambiguous which one would be better.

3. "Step #2". This consists of running over the points x_i, according to the order identified in number 1, and at each step, updating the "row" $c_{i,\cdot}$ while holding all other c entries constant. The possibility for different orders having different results, results from the fact that each subsequent step depends on the values in the column $c_{\cdot,n+1}$, some of which were previously updated and others which remain to be updated.

4. Recall that the entries in the column $c_{\cdot,n+1}$ represent what we actually want in the end. Therefore, a termination condition for this algorithm is whether the magnitude of the change in this column, after executing Step #2, is less than some tolerance ϵ. Otherwise, we repeat Step #2. Or maybe we run too many iterations, and have to quit.

For each i, Step #2 requires finding values for the row $c_{i,\cdot}$ such that $\mathcal{J}_i(c)$ is minimized. For most applications, the entries in c cannot take on any value; they must, for example, be constrained to be between 0 and 1 (inclusive). The paper solves this problem by taking the unconstrained minimum, and then setting to 0 all negative values and to 1 all values greater than 1. Additionally, finding this minimum in the form that $\mathcal{J}_i(c)$ is written is computationally expensive.

Let us consider the following. For each $i \in \{1, \ldots, N-2n\}$, suppose $\mathbf{U}, \mathbf{V}, \mathbf{W}, \mathbf{Z} \in \mathbb{R}^{2n}$ with:

$$\begin{aligned}
U_m &= \sum_{l=1}^{2n} \Delta_{i+m-1,l-i-2n+1} c_{i,l} \\
V_m &= \sum_{j=1}^{N-2n} \{\text{conditions}\} \Delta_{i+m-1,j+n} c_{j,n+1} \\
W_m &= f^{\delta}(x_i + \rho_m) \\
Z_m &= c_{i,m}
\end{aligned}$$

Then we can say:

$$\mathcal{J}_i(c) = \|\mathbf{U}_i + \mathbf{V}_i - \mathbf{W}_i\|^2 + \alpha(x_i)\|\mathbf{Z}_i\|^2$$

Additionally, it is clear that $\mathbf{U}_i, \mathbf{V}_i$ are matrix-vector multiplications. We let $\mathbf{A}^{(i)}, \mathbf{B}^{(i)}$ be those matrices, with entries clearly being:

$$\begin{aligned}
A^{(i)}_{m,l} &= \Delta_{i+m-1,l-i-2n+1} \\
B^{(i)}_{m,j} &= \{\text{conditions}\} \Delta_{i+m-1,j+n}
\end{aligned}$$

However, the vectors to which these are applied are not the same. $\mathbf{A}^{(i)}$ is applied to the row $c_{i,\cdot}$ which we shall denote $\mathbf{c}^{(i)}$, and $\mathbf{B}^{(i)}$ is applied to the column which we shall denote $\tilde{\mathbf{c}}$. Also note that $\mathbf{Z} = \mathbf{c}^{(i)}$. $\mathbf{A}^{(i)} \in \mathbb{R}^{2n \times 2n}, \mathbf{B}^{(i)} \in \mathbb{R}^{2n \times (N-2n)}$. So we want to minimize:

$$\|\mathbf{A}^{(i)}\mathbf{c}^{(i)} + \mathbf{B}^{(i)}\tilde{\mathbf{c}} - \mathbf{W}_i\|^2 + \alpha(x_i)\|\mathbf{c}^{(i)}\|^2$$

With respect to $\mathbf{c}^{(i)}$, keeping $\tilde{\mathbf{c}}$ constant. (Though these two do share an entry... is that not a problem?) Apparently, this is equivalent to minimizing:

$$\|\mathbf{D}^{(i)}\mathbf{c}^{(i)} - \mathbf{g}^{(i)}\|^2$$

Where

$$\mathbf{D}^{(i)} = \begin{pmatrix} \mathbf{A}^{(i)} \\ \sqrt{\alpha(x_i)}\mathbf{I} \end{pmatrix}, \quad \mathbf{g}^{(i)} = \begin{pmatrix} \mathbf{W}^{(i)} - \mathbf{B}^{(i)}\tilde{\mathbf{c}} \\ \mathbf{0} \end{pmatrix}$$

$$\mathbf{I} \in \mathbb{R}^{2n\times 2n}, \quad \mathbf{0} \in \mathbb{R}^{2n}$$

One approach to solving this problem is to solve the associated *normal equations*:

$$\mathbf{D}^{(i)\top}\mathbf{D}^{(i)}\mathbf{c}^{(i)} = \mathbf{D}^{(i)\top}\mathbf{g}^{(i)}$$

Note that the "normal matrix" $\mathbf{D}^{(i)\top}\mathbf{D}^{(i)} = \mathbf{A}^{(i)\top}\mathbf{A}^{(i)} + \alpha(x_i)\mathbf{I}$, and in the case of convolution kernels, $\mathbf{A}^{(i)}$ does not depend on i. For example, in our case,

$$\begin{aligned}
\Delta_{i+m-1,l-i-2n+1} &= \int_{(l-i-2n)/N}^{(l-i-2n+1)/N} k\left(\frac{i+m-2}{N}, s\right) \mathrm{d}s \\
&= \frac{1}{\sqrt{2\pi\sigma^2}} \int_{(l+m-2n-2)/N}^{(l+m-2n-1)/N} e^{-t^2/2\sigma^2} \mathrm{d}s \\
&= \frac{1}{2}\left(\operatorname{erf}\left(\frac{l+m-2n-1}{\sqrt{2}\sigma N}\right) - \operatorname{erf}\left(\frac{l+m-2n-2}{\sqrt{2}\sigma N}\right)\right)
\end{aligned}$$

Which depends on m, l, as expected, but not on i. We use the convention:

$$\operatorname{erf}(x) := \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \mathrm{d}t$$

which is consistent with MATLAB and Mathematica, but not universal.

In the case that it is a convolution kernel and α is constant, then the LU decomposition, or in fact the inverse, of $\mathbf{D}^{(i)\top}\mathbf{D}^{(i)}$ may be computed once at the beginning and then never again. If it is a convolution kernel and α does vary over the image, then the LU decomposition of $\mathbf{A}^{(i)\top}\mathbf{A}^{(i)}$ may be stored, and the rest computed more easily at each step. Conceivably, α may only take on a few different values over the entire image, and then it might be possible to store the LU, or inverse, for each of these few values.

For $\mathbf{B}^{(i)}$, we can similarly compute the coefficients in our case.

$$\Delta_{i+m-1,j+n} = \frac{1}{2}\left(\operatorname{erf}\left(\frac{i+m-j-n-1}{\sqrt{2}\sigma N}\right) + \operatorname{erf}\left(\frac{i+m-j-n-2}{\sqrt{2}\sigma N}\right)\right)$$

3 Other Ideas

Without piecewise constant constraint?