IRLS nuclear norm minimization for FRI annihilating filter method

Greg Ongie

October 11, 2015

1 IRLS algorithm

Our goal is to recover the k-space data $\hat{f}$ on a rectangular Cartesian grid $\Delta \subseteq \mathbb{Z}^2$ from undersampled measurements $P_\Gamma \hat{f} = \mathbf{b}$, where P_Γ is the projection onto the sampling set $\Gamma \subset Z^2$. We use the low-rank property of the annihilation matrix as a regularization prior on the data. This may be posed as

$$\min_{\hat{f}} \|\mathcal{T}(\hat{f})\|_* + \lambda \|P_\Gamma \hat{f} - \mathbf{b}\|_2^2 \tag{1}$$

where here $\mathcal{T}(\hat{f}) \in \mathbb{C}^{2M \times N}$ is a large structured matrix built from $\hat{f}$ representing the system of annihilation equations. Here M is approximately the product of the dimensions of the reconstruction grid, and N is the product of the dimensions of the annihilating filter. I will give more details on the structure of $\mathcal{T}(\hat{f})$ in the next section.

We will solve (1) using the IRLS algorithm, which results in a series of quadratic sub-problems:

$$\hat{f}_{n+1} = \arg\min_{\hat{f}} \|\mathcal{T}(\hat{f})\mathbf{W}_n^{1/2}\|_F^2 + \lambda \|P_\Gamma \hat{f} - \mathbf{b}\|_2^2 \tag{2}$$

where the *weight matrix* W_n is updated in every iterate according to

$$\mathbf{W}_n = [\mathcal{T}(\hat{f}_n)^* \mathcal{T}(\hat{f}_n)]^{-1/2}, \tag{3}$$

Note that (2) may be solved with CG, which should be fast, but W_n is updated using an SVD, which is the computational bottleneck here. The hope is that a decent solution can be obtained after only ≈10-20 iterates, rather than the ≈100 SVD's the singular value thresholding algorithm would require.

2 Simplifications

Now I will show how the iterates in (2) and (3) simplify in our setting, and how the scheme may be interpreted as alternating between enforcing an annihilation constraint and estimation of an annihilating filter.

2.1 Structure of $\mathcal{T}(\hat{f})$

Given any k-space data $\hat{g}$ on all of Δ, we define $\mathcal{T}(\hat{g})$ as

$$\mathcal{T}(\hat{g}) = \begin{bmatrix} \mathbf{T}_x \\ \mathbf{T}_y \end{bmatrix}$$

where both $\mathbf{T}_x$ and $\mathbf{T}_y$ are block Toeplitz matrices (with Toeplitz blocks) which can be thought of as representing 2-D convolution with $\widehat{\partial_x f} = -j\omega_x \hat{f}$ and $\widehat{\partial_x f} = -j\omega_y \hat{f}$. That is, if $\mathbf{c}$ is any (vectorized) filter with indices in $\Lambda \subseteq \mathbb{Z}^2$, then $\mathbf{T}_x$ and $\mathbf{T}_y$ are defined by

$$\mathbf{T}_x \mathbf{c} = P(\widehat{\partial_x f} * \mathbf{c})$$
$$\mathbf{T}_y \mathbf{c} = P(\widehat{\partial_y f} * \mathbf{c})$$

where P is projection onto indices where the convolution is valid, namely, the set $\Delta|\Lambda = \{l \in \Delta : k - l \in \Delta \text{ for all } k \in \Lambda\}$.

2.2 Step one: Least-squares annihilation

First, we focus on solving (2). Consider the more general case where in place of $\mathbf{W}_n$ we have an arbitrary filterbank $\mathbf{D} = [\mathbf{d}_1, ..., \mathbf{d}_n]$, i.e. each one of the columns $\mathbf{d}_i$ represents a filter, and we wish to solve for k-space data $\hat{f}$ which is best annihilated by each filter $\mathbf{d}_i$ in a least squares sense, subject to data constraints. Then we may solve

$$\min_{\hat{f}} \|\mathcal{T}(\hat{f})\mathbf{D}\|_F^2 + \lambda \|P_\Gamma \hat{f} - \mathbf{b}\|_2^2$$

or, equivalently

$$\min_{\hat{f}} \sum_{i=1}^{m} \|\mathcal{T}(\hat{f})\mathbf{d}_i\|_2^2 + \lambda \|P_\Gamma \hat{f} - \mathbf{b}\|_2^2$$

To help us find the gradient of the above cost function, define the linear operator $\mathcal{Q}_i$ by $\mathcal{Q}_i \hat{f} = \mathcal{T}(\hat{f})\mathbf{d}_i = P(\mathbf{d}_i * \widehat{\nabla f})$, for all $I = 1, ..., n$. We may expand each $\mathcal{Q}_i$ into a series of linear operators

$$\mathcal{Q}_i = \mathbf{P}\mathbf{C}_i\mathbf{M}$$

Here $\mathbf{M}$ is element-wise multiplication by $-j\boldsymbol{\omega}$, $\mathbf{C}_i$ is convolution by the filter $\mathbf{d}_i$, and $\mathbf{P}$ is projection onto the set of valid convolution indices. Because of the projection, we may assume $\mathbf{C}_i$ is a circular convolution (provided we zero-pad things sufficiently), and so we may write $\mathbf{C}_i = \mathbf{F}\mathbf{S}_i\mathbf{F}^*$ where $\mathbf{F}$ is the 2-D DFT on a large grid, and $\mathbf{S}_i$ is a diagonal matrix representing pointwise multiplication by the inverse DFT of the filter $\mathbf{d}_i$.

In this notation, the problem is now to solve

$$\min_{\hat{f}} \sum_{i=1}^{m} \|\mathcal{Q}_i \hat{f}\|_2^2 + \lambda \|P_\Gamma \hat{f} - \mathbf{b}\|_2^2,$$

Computing the gradient of the objective and setting it equal to zero yields the linear system:

$$\underbrace{\left(\sum_{i=1}^{m} \mathcal{Q}_i^* \mathcal{Q}_i + \lambda P_\Gamma^* P_\Gamma\right)}_{\mathbf{R}} \hat{f} = \lambda P_\Gamma^* b$$

and we have

$$\mathcal{Q}_i^* \mathcal{Q}_i = \mathbf{M}^* \mathbf{C}_i^* \mathbf{P}^* \mathbf{P} \mathbf{C}_i \mathbf{M}.$$

If the reconstruction grid is large relative to the filter size, the projection operator $\mathbf{P}^*\mathbf{P}$ will be very close to identity. Hence, we propose making the approximation $\mathbf{P}^*\mathbf{P} \approx \mathbf{I}$, so that the above becomes

$$\mathcal{Q}_i^* \mathcal{Q}_i \approx \mathbf{M}^* \mathbf{C}_i^* \mathbf{C}_i \mathbf{M} = \mathbf{M}^* \mathbf{F} |\mathbf{S}_i|^2 \mathbf{F}^* \mathbf{M}$$

And so

$$\mathbf{R} = \mathbf{M}^* \mathbf{F} \left(\sum_{i=1}^{m} |\mathbf{S}_i|^2\right) \mathbf{F}^* \mathbf{M} + \lambda \mathbf{P}_\Gamma^* \mathbf{P}_\Gamma$$

Therefore to apply $\mathbf{R}$ we only need to compute $\sum_i |\mathbf{S}_i|^2$, which can be thought of as multiplication with a gridded version of the sum-of-squares polynomial

$$\overline{\mu}(\mathbf{r}) = \sum_{i=1}^{m} |\mu_i(\mathbf{r})|^2. \tag{4}$$

where $\mu_i(\mathbf{r})$ is the inverse Fourier transform of $\mathbf{d}_i$, i.e. the trigonometric polynomial having coefficients $\mathbf{d}_i[\mathbf{k}], \mathbf{k} \in \Lambda$.

2.3 Step Two: Annihilating Mask Update

Now consider step 2 of the algorithm, the weight matrix update (3). Recall that $\mathbf{W}_n = [\mathcal{T}(f_n)^* \mathcal{T}(f_n)]^{-1/2}$. Therefore, according to the previous section, the first step of the algorithm (2) becomes a least squares annihilation where the filter bank is specified by $\mathbf{W}_n^{1/2} = [\mathbf{w}_1, ..., \mathbf{w}_N]$.

Let $\mathcal{T}(f_n) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$ be the SVD. Then $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_0]$ where $\mathbf{V}_0$ is a basis of the null space of $\mathcal{T}(f_n)$, and hence a basis of annihilating filters. From $\mathbf{W} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{V}^*$, we have $\mathbf{W}^{1/2} = \mathbf{V}\mathbf{\Sigma}^{-1/2} = [\sigma_1^{-1/2}\mathbf{v}_1, \ldots, \sigma_n^{-1/2}\mathbf{v}_N]$. But step one of the algorithm only needs the sum-of-squares polynomial (4) defined by the filterbank, which in this case can be expressed as

$$\overline{\mu}(\mathbf{r}) = \sum_{i=1}^{N} \sigma_i^{-1} |\gamma_i(\mathbf{r})|^2 \tag{5}$$

where γ_i is the inverse Fourier transform of the filter $\mathbf{v}_i$. Note σ_i^{-1} is small when corresponding filter is in the co-kernel (i.e. the "signal subspace") and will be large when it is in the nullspace $\mathbf{V}_n$ (i.e. when it is an annihilating filter). Therefore, we may interpret $\overline{\mu}$ as a regularized annihilating filter.

Finally, note that in practice we actually use the ϵ-regularized inverse on the singular values in (5), i.e. instead of using σ^{-1}, which is undefined for $\sigma = 0$, we use

$$\sigma_\epsilon^{-1} := \frac{1}{\max(\sigma, \epsilon)}$$

or something similar. Typically, there is also a rule for shrinking $\epsilon \to 0$ as the iterates proceed, which speeds up the convergence.

2.4 Alternative mask updates

Rather than forming the mask according to (5), which weights the singular vectors by σ^{-1} (or σ_ϵ^{-1}), we can consider a general update of the form

$$\overline{\mu}(\mathbf{r}) = \sum_{i=1}^{N} \varphi(\sigma_i)|\gamma_i(\mathbf{r})|^2 \tag{6}$$

Where $\varphi : \mathbb{R}^+ \to \mathbb{R}^+$ is any weighting function which is large for small inputs. For example, for any $q > 0$ we could choose

$$\varphi(t) = t^{-q}$$

which generalizes the update (5). Choosing $q > 1$ could potentially work better, since this would further shrink filters corresponding to high singular values, and amplify filters in the null space, as desired. I did some experiments with $q = 2$, and it shows superior performance to $q = 1$. I believe this weighting corresponds replacing the nuclear norm with a Shatten p-norm for $p = 1/q < 1$, but I need to check the details.

Another choice would be the hard threshold

$$\varphi(t) = \begin{cases} 1 & \text{if } t < T \\ 0 & \text{else,} \end{cases}$$

for some $T > 0$, but this is not likely to be stable.

Additionally, we could consider truncate the sum in (6), as

$$\overline{\mu}(\mathbf{r}) = \sum_{i=R}^{N} \varphi(\sigma_i)|\gamma_i(\mathbf{r})|^2 \tag{7}$$

where R is some estimate on a lower-bound of the true rank. This way, we give zero weight to the first R filters corresponding to the highest singular values, which are not annihilating filters, and should not be incorporated into the mask. This showed good performance in experiments, too. I believe this corresponds to a weighting that would result from a LORAKS-type penalty in the original objective rather than the nuclear norm.