Chambolle–Pock algorithm for solving the L+S model

Ondřej Mokrý

December 20, 2022

1 Introduction and general form of CPA

The goal is to find the components $\mathbf{L}, \mathbf{S}$, such that it fits the measured data $\mathbf{d}$, formally:

$$\hat{\mathbf{L}}, \hat{\mathbf{S}} = \arg\min_{\mathbf{L},\mathbf{S}} \frac{1}{2} \left\| \mathbf{d} - \mathcal{A}(\mathbf{L}+\mathbf{S}) \right\|_2^2 + \lambda_L \left\| \mathbf{L} \right\|_* + \lambda_S \left\| T\mathbf{S} \right\|_1 , \tag{1}$$

where the operator $\mathcal{A}$ represents the acquisition process. The parameters $\lambda_L, \lambda_S > 0$ control the trade-off between the data fidelity, low-rankness and sparsity of the solution $\hat{\mathbf{x}} = \hat{\mathbf{L}} + \hat{\mathbf{S}}$.

Since the objective function of (1) is non-smooth, we propose to approach it with a suitable proximal method [1]. Most of the simple algorithms are designed to minimize a sum of two functions, and the Chambolle–Pock (primal–dual) proximal algorithm (CPA) [2] can manage also a composition with a linear operator, such as T in Eq. (1). To apply the algorithm, we rearrange the problem into the following form:

$$\hat{\mathbf{L}}, \hat{\mathbf{S}} = \arg\min_{\mathbf{L},\mathbf{S}} f\left(K \begin{bmatrix} \mathbf{L} \\ \mathbf{S} \end{bmatrix} \right) + g\left(\begin{bmatrix} \mathbf{L} \\ \mathbf{S} \end{bmatrix} \right), \tag{2}$$

where

$$f\left(\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} \right) = \frac{1}{2} \left\| \mathbf{d} - \mathbf{u} \right\|_2^2 + \lambda_S \left\| \mathbf{v} \right\|_1 , \quad g\left(\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} \right) = \lambda_L \left\| \mathbf{u} \right\|_* , \quad K = \begin{bmatrix} \mathcal{A} & \mathcal{A} \\ 0 & T \end{bmatrix}. \tag{3}$$

A single iteration of the algorithm, given the parameters σ, τ such that $\sigma\tau \left\| K \right\|^2 < 1$, is then performed as

$$\mathbf{y}^{(k+1)} = \mathrm{prox}_{\sigma f^*}\left(\mathbf{y}^{(k)} + \sigma K \bar{\mathbf{x}}^{(k)} \right), \tag{4a}$$

$$\mathbf{x}^{(k+1)} = \mathrm{prox}_{\tau g}\left(\mathbf{x}^{(k)} - \tau K^* \mathbf{y}^{(k+1)} \right), \tag{4b}$$

$$\bar{\mathbf{x}}^{(k+1)} = 2\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}. \tag{4c}$$

The main variable $\mathbf{x}$ is the concatenation of $\mathbf{L}$ and $\mathbf{S}$ from Eq. (2) and the dual variable $\mathbf{y}$ belongs in the domain of the (adjoint) operator K^*.

2 The proximal operators

First, recall the scaled Moreau identity: Given a convex function f, the proximal operator of its Fenchel–Rockafellar conjugate [3] f^* can be computed as [4, 5]

$$\mathrm{prox}_{\alpha f^*}(\mathbf{u}) = \mathbf{u} - \alpha \, \mathrm{prox}_{f/\alpha}(\mathbf{u}/\alpha) \quad \text{for } \alpha \in \mathbb{R}^+. \tag{5}$$

Using the separability of f and g of Eq. (3), the proximal operators can be decomposed [6, Thm. 6.6] end expressed [1, Tab. I] as

$$\operatorname{prox}_{\eta f}\left(\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}\right)=\begin{bmatrix}\operatorname{prox}_{\frac{\eta}{2}\|\mathbf{d}-\cdot\|_2^2}(\mathbf{u})\\ \operatorname{prox}_{\eta\lambda_S\|\cdot\|_1}(\mathbf{v})\end{bmatrix}=\begin{bmatrix}\frac{\mathbf{u}+\eta\mathbf{d}}{1+\eta}\\ \operatorname{soft}_{\eta\lambda_S}(\mathbf{v})\end{bmatrix}, \tag{6a}$$

$$\operatorname{prox}_{\tau g}\left(\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}\right)=\begin{bmatrix}\operatorname{prox}_{\tau\lambda_L\|\cdot\|_*}(\mathbf{u})\\ \operatorname{prox}_0(\mathbf{v})\end{bmatrix}=\begin{bmatrix}\operatorname{SVT}_{\tau\lambda_L}(\mathbf{u})\\ \mathbf{v}\end{bmatrix}. \tag{6b}$$

The operator $\operatorname{soft}_\alpha$ denotes soft thresholding with threshold α defined as

$$\operatorname{soft}_\alpha(\mathbf{u})=\operatorname{sign}(\mathbf{u})\cdot\max(|\mathbf{u}|-\alpha,0). \tag{7}$$

The singular value thresholding [7] with threshold α, denoted $\operatorname{SVT}_\alpha$, is defined using the singular value decomposition $\mathbf{u}=\mathbf{U}\operatorname{diag}(\mathbf{s})\mathbf{V}^{\mathrm{H}}$ as

$$\operatorname{SVT}_\alpha(\mathbf{u})=\mathbf{U}\operatorname{diag}\left(\operatorname{soft}_\alpha(\mathbf{s})\right)\mathbf{V}^{\mathrm{H}}. \tag{8}$$

Finally, we expand the proximal operator $\operatorname{prox}_{\sigma f^*}$ using Eq. (5) and (6a) as

$$\operatorname{prox}_{\sigma f^*}\left(\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}\right)=\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}-\sigma\begin{bmatrix}\frac{\mathbf{u}/\sigma+\mathbf{d}/\sigma}{1+1/\sigma}\\ \operatorname{soft}_{\lambda_S/\sigma}(\mathbf{v}/\sigma)\end{bmatrix}=\begin{bmatrix}\mathbf{u}\\ \mathbf{v}\end{bmatrix}-\begin{bmatrix}\frac{\mathbf{u}+\mathbf{d}}{1+1/\sigma}\\ \operatorname{soft}_{\lambda_S}(\mathbf{v})\end{bmatrix}=\begin{bmatrix}\frac{\mathbf{u}-\sigma\mathbf{d}}{1+\sigma}\\ \mathbf{v}-\operatorname{soft}_{\lambda_S}(\mathbf{v})\end{bmatrix}. \tag{9}$$

3 Particular form of CPA

Denote

$$\mathbf{y}^{(k)}=\begin{bmatrix}\mathbf{M}^{(k)}\\ \mathbf{N}^{(k)}\end{bmatrix},\quad \mathbf{x}^{(k)}=\begin{bmatrix}\mathbf{L}^{(k)}\\ \mathbf{S}^{(k)}\end{bmatrix},\quad \bar{\mathbf{x}}^{(k)}=\begin{bmatrix}\bar{\mathbf{L}}^{(k)}\\ \bar{\mathbf{S}}^{(k)}\end{bmatrix}, \tag{10}$$

and recall from Eq. (3) that

$$K=\begin{bmatrix}\mathcal{A} & \mathcal{A}\\ 0 & T\end{bmatrix},\quad K^*=\begin{bmatrix}\mathcal{A}^* & 0\\ \mathcal{A}^* & T^\top\end{bmatrix}. \tag{11}$$

For the time difference operator T, we use the non-hermitian transpose, since it is real. Then,

$$\texttt{argf}=\mathbf{y}^{(k)}+\sigma K\bar{\mathbf{x}}^{(k)}=\begin{bmatrix}\mathbf{M}^{(k)}+\sigma\mathcal{A}(\bar{\mathbf{L}}^{(k)}+\bar{\mathbf{S}}^{(k)})\\ \mathbf{N}^{(k)}+\sigma T\bar{\mathbf{S}}^{(k)}\end{bmatrix}, \tag{12a}$$

$$\texttt{argg}=\mathbf{x}^{(k)}-\tau K^*\mathbf{y}^{(k+1)}=\begin{bmatrix}\mathbf{L}^{(k)}-\tau\mathcal{A}^*\mathbf{M}^{(k+1)}\\ \mathbf{S}^{(k)}-\tau\mathcal{A}^*\mathbf{M}^{(k+1)}-\tau T^\top\mathbf{N}^{(k+1)}\end{bmatrix}. \tag{12b}$$

The CPA iteration (4) then attains the form

$$\mathbf{M}^{(k+1)}=\left(\mathbf{M}^{(k)}+\sigma\mathcal{A}(\bar{\mathbf{L}}^{(k)}+\bar{\mathbf{S}}^{(k)})-\sigma\mathbf{d}\right)/\left(1+\sigma\right), \tag{13a}$$

$$\mathbf{N}^{(k+1)}=\mathbf{N}^{(k)}+\sigma T\bar{\mathbf{S}}^{(k)}-\operatorname{soft}_{\lambda_S}\left(\mathbf{N}^{(k)}+\sigma T\bar{\mathbf{S}}^{(k)}\right), \tag{13b}$$

$$\mathbf{L}^{(k+1)}=\operatorname{SVT}_{\tau\lambda_L}\left(\mathbf{L}^{(k)}-\tau\mathcal{A}^*\mathbf{M}^{(k+1)}\right), \tag{13c}$$

$$\mathbf{S}^{(k+1)}=\mathbf{S}^{(k)}-\tau\mathcal{A}^*\mathbf{M}^{(k+1)}-\tau T^\top\mathbf{N}^{(k+1)}, \tag{13d}$$

$$\bar{\mathbf{L}}^{(k+1)}=2\mathbf{L}^{(k+1)}-\mathbf{L}^{(k)}, \tag{13e}$$

$$\bar{\mathbf{S}}^{(k+1)}=2\mathbf{S}^{(k+1)}-\mathbf{S}^{(k)}. \tag{13f}$$

The values $\sigma, \tau > 0$ should be chosen such that $\sigma\tau \|K\|^2 < 1$, i.e. $\sigma\tau < 1/\|K\|^2$. It holds

$$\|K\|^2 = \|K^*K\| = \left\| \begin{bmatrix} \mathcal{A}^* & 0 \\ \mathcal{A}^* & T^\top \end{bmatrix} \begin{bmatrix} \mathcal{A} & \mathcal{A} \\ 0 & T \end{bmatrix} \right\| = \left\| \begin{bmatrix} \mathcal{A}^*\mathcal{A} & \mathcal{A}^*\mathcal{A} \\ \mathcal{A}^*\mathcal{A} & \mathcal{A}^*\mathcal{A} + T^\top T \end{bmatrix} \right\|$$
$$\leq 3\|\mathcal{A}^*\mathcal{A}\| + \|\mathcal{A}^*\mathcal{A} + T^\top T\| \leq 4\|\mathcal{A}^*\mathcal{A}\| + \|T^\top T\| = 4\|\mathcal{A}\|^2 + \|T\|^2. \quad (14)$$

Therefore, it is sufficient to bound the product $\sigma\tau$ with the threshold $1/(4\|\mathcal{A}\|^2 + \|T\|^2) \leq 1/\|K\|^2$.

4 Possible parametrizations

4.1 All the scalar parameters

$$\begin{aligned}
\mathbf{M}^{(k+1)} &= \left(\mathbf{M}^{(k)} + \sigma_1^{(k)} \mathcal{A}(\bar{\mathbf{L}}^{(k)} + \bar{\mathbf{S}}^{(k)}) - \sigma_2^{(k)} \mathbf{d}\right) / \left(1 + \sigma_3^{(k)}\right) \\
\mathbf{N}^{(k+1)} &= \mathbf{N}^{(k)} + \sigma_4^{(k)} T\bar{\mathbf{S}}^{(k)} - \operatorname{soft}_{\lambda_S^{(k)}} \left(\mathbf{N}^{(k)} + \sigma_5^{(k)} T\bar{\mathbf{S}}^{(k)}\right) \\
\mathbf{L}^{(k+1)} &= \operatorname{SVT}_{\tau_2^{(k)} \lambda_L^{(k)}} \left(\mathbf{L}^{(k)} - \tau_1^{(k)} \mathcal{A}^* \mathbf{M}^{(k+1)}\right) \\
\mathbf{S}^{(k+1)} &= \mathbf{S}^{(k)} - \tau_3^{(k)} \mathcal{A}^* \mathbf{M}^{(k+1)} - \tau_4^{(k)} T^\top \mathbf{N}^{(k+1)} \\
\bar{\mathbf{L}}^{(k+1)} &= 2\mathbf{L}^{(k+1)} - \mathbf{L}^{(k)} \\
\bar{\mathbf{S}}^{(k+1)} &= 2\mathbf{S}^{(k+1)} - \mathbf{S}^{(k)}
\end{aligned}$$

4.2 APL

The thresholding (both $\operatorname{soft}_{\lambda_S}$ applied on the time differences and $\operatorname{soft}_{\tau\lambda_L}$ inside the $\operatorname{SVT}_{\tau\lambda_L}$) can be replaced by the Adaptive piecewise-linear activations (APL). APL has the form [8]

$$h(x) = \max(0, x) + \sum_{s=1}^{S} a_s \max(0, -x + b_s), \quad (16)$$

which can be rewritten using $\operatorname{ReLU}(x) = \max(0, x)$ as

$$h(x) = \operatorname{ReLU}(x) + \sum_{s=1}^{S} a_s \operatorname{ReLU}(-x + b_s). \quad (17)$$

If $a_s < 0$, $s = 1, \dots, S$, the scalar function h is non-decreasing, thus it is a proximal operator of some (not necessarily convex) penalty [9, Corollary 7]. This is directly extended to separable functions [6, Theorem 6.6].

Regarding complex functions, APL can be applied either separately on the real and imaginary parts, or it can work with the magnitude while keeping the phase of the argument. To analyze the latter approach, note that to determine the proximal operator for a function that depends only on the magnitude of its argument, we simply determine the proximal operator for the magnitude and apply the sign of the argument at the end. To prove this, denote $f(z) = g(|z|)$, $f\colon \mathbb{C} \to \mathbb{R}$,

$g\colon \mathbb{R}_+ \to \mathbb{R}$. Then[1]

$$\begin{aligned}\operatorname{prox}_f(v) &= \arg\min_{z\in\mathbb{C}} \frac{1}{2}|v-z|^2 + f(z)\\ &= \arg\min_{m\mathrm{e}^{\mathrm{i}\phi}: m\geq 0, \phi\in\mathbb{R}} \frac{1}{2}|v - m\mathrm{e}^{\mathrm{i}\phi}|^2 + f(m\mathrm{e}^{\mathrm{i}\phi})\\ &= \arg\min_{m\mathrm{e}^{\mathrm{i}\phi}: m\geq 0, \phi\in\mathbb{R}} \frac{1}{2}\left||v|\mathrm{e}^{\mathrm{i}\angle v} - m\mathrm{e}^{\mathrm{i}\phi}\right|^2 + g(m).\end{aligned}$$

Since the quadratic term is minimized with respect to the angle ϕ at $\phi_* = \angle v$, we proceed to optimize for m:

$$m_* = \arg\min_{m\geq 0} \frac{1}{2}\left||v| - m\right|^2 + g(m) = \operatorname{prox}_g(|v|), \tag{18}$$

where we take into account that the function g is defined only for non-negative arguments, or, equivalently, it attains infinite value for negative arguments. To summarize,

$$\operatorname{prox}_f(v) = \operatorname{prox}_g(|v|)\mathrm{e}^{\mathrm{i}\angle v}. \tag{19}$$

Therefore, the complex APL working with the magnitude only can still be a proximal operator. However, the constraint $a_s < 0$, $s = 1, \ldots, S$ is not sufficient, since the optimal magnitude in Eq. (18) needs to be non-negative.

References

[1] P. Combettes and J. Pesquet, "Proximal splitting methods in signal processing," *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, vol. 49, pp. 185–212, 2011.

[2] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[3] S. P. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[4] N. Komodakis and J. Pesquet, "Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems," *IEEE Signal Processing Magazine*, vol. 32, pp. 31–54, Nov. 2015.

[5] L. Condat, "A generic proximal algorithm for convex optimization—application to total variation minimization," *Signal Processing Letters, IEEE*, vol. 21, pp. 985–989, Aug. 2014.

[6] A. Beck, *First-Order Methods in Optimization.* SIAM-Society for Industrial and Applied Mathematics, 2017.

[7] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, pp. 1956–1982, jan 2010.

[8] F. Agostinelli, M. Hoffman, P. Sadowski, and P. Baldi, "Learning activation functions to improve deep neural networks," 2014.

[9] R. Gribonval and M. Nikolova, "A characterization of proximity operators," *Journal of Mathematical Imaging and Vision*, vol. 62, pp. 773–789, July 2020.

[1] https://web.eecs.umich.edu/~fessler/course/598/l/n-05-prox.pdf