S2 Text: Details about PADMM

The PADMM is a variation of the alternating direction method of multipliers. Before introducing the PADMM, we present the ADMM algorithm for solving these three subproblems. The scaled ADMM is employed here since it is a more convenient form [1].

For subproblem 1, first, we convert it into ADMM form:

\[
\min_{A(1), B(1), D(1)} \frac{1}{2} \|A(1)Y(1) - [Y(2) - Y(1)]\|_F^2 + \alpha\|B(1)\|_1 + \beta\|A(2)^k - D(1)\|_1
\]

s.t. \(B(1) - A(1) = 0, D(1) - A(1) = 0\).

Its augmented Lagrangian is

\[
L_\rho(A(1), B(1), D(1), U(1), W(1))
= \frac{1}{2} \|A(1)Y(1) - [Y(2) - Y(1)]\|_F^2 + \alpha\|B(1)\|_1 + \beta\|A(2)^k - D(1)\|_1
+ \frac{\rho}{2} \|B(1) - A(1) + U(1)\|_F^2 - \frac{\rho}{2} \|U(1)\|_F^2 + \rho \|D(1) - A(1) + W(1)\|_F^2
- \frac{\rho}{2} \|W(1)\|_F^2.
\]

The iterations are as follows:

\[
\begin{align*}
(A(1))^{k+1} &= [(Y(2) - Y(1)) \cdot Y(1)^T + \rho^k (B(1)^k + U(1)^k + D(1)^k + W(1)^k)] \cdot [Y(1)Y(1)^T + 2 \rho^k I]^{-1}
B(1))^{k+1} &= S_{\alpha/\rho^k}(A(1)^k - U(1)^k)
U(1))^{k+1} &= U(1)^k + B(1)^k - A(1)^k
D(1))^{k+1} &= S_{\beta/\rho^k}(A(1)^k - W(1)^k - A(2)^k) + A(2)^k
W(1))^{k+1} &= W(1)^k + D(1)^k - A(1)^k,
\end{align*}
\]

where the soft thresholding operator \(S\) is defined as

\[
S_\kappa(a) = \begin{cases} 
    a - \kappa & a > \kappa \\
    0 & |a| \leq \kappa \\
    a + \kappa & a < -\kappa
\end{cases}
\]

For subproblem 2, we convert it into ADMM form:

\[
\min_{A(t), B(t), C(t), D(t)} \frac{1}{2} \|A(t)Y(t) - [Y(t + 1) - Y(t)]\|_F^2 + \alpha\|B(t)\|_1
+ \beta\|A(t + 1) - C(t)\|_1 + \beta\|D(t) - A(t - 1)\|_1
\]

s.t. \(B(t) - A(t) = 0, C(t) - A(t) = 0, D(t) - A(t) = 0\).

Its augmented Lagrangian is
Usually, in the ADMM, the update to PADMM is used to achieve a faster computation speed. Proper iterations are as follows:

\[
\begin{align*}
\frac{A(t)^{k+1}}{(A(t)^{k+1})} &= [(Y(t + 1) - Y(t)) \cdot Y(t)^T + \rho^k (B(t)^k + U(t)^k + C(t)^k + V(t)^k + D(t)^k + W(t)^k)] \cdot [Y(t)^T Y(t)^T + 3\rho^k I]^{-1} \\
\frac{B(t)^{k+1}}{(B(t)^{k+1})} &= S_{\nu(p^k)}(A(t)^{k+1}) - U(t)^k \\
\frac{U(t)^{k+1}}{(U(t)^{k+1})} &= U(t)^k + B(t)^k - A(t)^k \\
\frac{C(t)^{k+1}}{(C(t)^{k+1})} &= S_{\nu(p^k)}(A(t)^{k+1}) - V(t)^k + A(t - 1)^k + A(t - 1)^k \\
\frac{V(t)^{k+1}}{(V(t)^{k+1})} &= V(t)^k + C(t)^k - A(t)^k \\
\frac{D(t)^{k+1}}{(D(t)^{k+1})} &= S_{\nu(p^k)}(A(t)^{k+1}) - W(t)^k + A(t + 1)^k + A(t + 1)^k \\
\frac{W(t)^{k+1}}{(W(t)^{k+1})} &= W(t)^k + D(t)^k - A(t)^k
\end{align*}
\]

Subproblem 3 is similar to subproblem 1. When \( t = N - 1 \),

\[
\begin{align*}
\frac{(A(N-1))^{k+1}}{(A(N-1)^{k+1})} &= [(Y(N) - Y(N-1)) \cdot Y(N-1)^T + \rho^k (B(N-1)^k + U(N-1)^k + C(N-1) + V(N-1)^k) \cdot [Y(N-1)^T Y(N-1)^T + 2\rho^k I]^{-1} \\
\frac{(B(N-1))^{k+1}}{(B(N-1)^{k+1})} &= S_{\nu(p^k)}(A(N-1)^{k+1}) - U(N-1)^k \\
\frac{(U(N-1))^{k+1}}{(U(N-1)^{k+1})} &= U(N-1)^k + B(N-1)^k - A(N-1)^k \\
\frac{(C(N-1))^{k+1}}{(C(N-1)^{k+1})} &= S_{\nu(p^k)}(A(N-1)^{k+1}) - V(N-1)^k + A(N-2)^k + A(N-2)^k \\
\frac{(V(N-1))^{k+1}}{(V(N-1)^{k+1})} &= V(N-1)^k + C(N-1)^k - A(N-1)^k
\end{align*}
\]

With some adjustments to the ADMM described above, one can use the PADMM to achieve a faster computation speed. Proper preconditioning processes are applied for the computation of \( A(t), 1 \leq t \leq N - 1 \).

The form of the iterations of \( A(t), 1 \leq t \leq N - 1 \), arises from \( \frac{\partial L_\rho}{\partial A(t)} = 0 \). Consider \( t = 1 \) (subproblem 1) as an example.

\[
\frac{\partial L_\rho(A(1), B(1), D(1), U(1), W(1))}{\partial A(1)} = 0
\]

\[
\begin{align*}
\frac{\partial L_\rho(A(1), B(1), D(1), U(1), W(1))}{\partial A(1)} &= A(1) \cdot [Y(1)Y(1)^T + 2\rho I] - (Y(2) - Y(1)) \cdot Y(1)^T - \rho(B(1) + U(1) + D(1) + W(1)) = 0
\end{align*}
\]

is equivalent to

\[
A(1) \cdot [Y(1)Y(1)^T + 2\rho I] = (Y(2) - Y(1)) \cdot Y(1)^T + \rho(B(1) + U(1) + D(1) + W(1)).
\]

Usually, in the ADMM, the update \( A(1) \) of takes the form

\[
A(1) = [(Y(2) - Y(1)) \cdot Y(1)^T + \rho(B(1) + U(1) + D(1) + W(1))] \cdot [Y(1)Y(1)^T + 2\rho I]^{-1}.
\]
With the proposed preconditioning, we add $-2\rho A(1)$ to both sides of Eq.\ref{eq11}. As the result,

$$A(1) = [(Y(2) - Y(1)) \cdot Y(1)^T + \rho(B(1) + U(1) + D(1) + W(1) - 2A(1))] \cdot [Y(1)Y(1)^T]^+,$$

where $(M)^+$ denotes the general inverse of the matrix $M$, in case $M$ is singular.

Similarly, we can obtain the PADMM iterations of $A(t), 1 \leq t \leq N - 1$ for all subproblems as follows:

$$A(t)^{t+1} = \begin{cases} [(Y(t + 1) - Y(t)) \cdot Y(t)^T + \rho(B(t)^2 + U(t)^2 + D(t)^2 + W(t)^2 - 2A(t)^2)] \cdot [Y(t)Y(t)^T]^+, & t = 1 \\ [(Y(t + 1) - Y(t)) \cdot Y(t)^T + \rho(B(t)^2 + U(t)^2 + C(t)^2 + V(t)^2 + D(t)^2 + W(t)^2 - 3A(t)^2)] \cdot [Y(t)Y(t)^T]^+, & 1 < t < N \\ [(Y(t - 1) - Y(t)) \cdot Y(t - 1)^T + \rho(B(t - 1)^2 + U(t - 1)^2 + C(t - 1)^2 + V(t - 1)^2 + D(t - 1)^2 + W(t - 1)^2 - 2A(t - 1)^2)] \cdot [Y(t - 1)Y(t - 1)^T]^+, & t = N \end{cases}$$

The $[Y(t)Y(t)^T]^+, 1 \leq t \leq N$, are unchanged in all iterations; therefore, they can be stored as constants. Hence, the PADMM can save $N$ matrix inversion computations in every iteration except the first. Singular value decomposition is used to compute the general inverses of the $[Y(t)Y(t)^T]^+, 1 \leq t \leq N$. Proper preconditioning makes the computation of the matrix inverses easier while maintaining an equivalent precision. Details on the theoretical results can be found in [2].
