MA3113: Topics in Mathematical Image Processing I
Sparse Representation and Dictionary Learning

Suh-Yuh Yang (楊肅煜)

Department of Mathematics, National Central University
Jhongli District, Taoyuan City 320317, Taiwan

First version: April 18, 2022/Last updated: April 15, 2024

Outlines

1. Sparse representation and dictionary learning
   - Sparse representation (SR) problem
   - Alternating direction method of multipliers (ADMM)
   - Application to signal denoising
   - Sparse dictionary learning (SDL) problem
   - Solving SDL problem
2. Convolutional sparse representation and dictionary learning
   - ADMM for solving convolutional SR problem
   - Convolutional SDL problem
   - Solving convolutional SDL problem

Part I

Sparse Representation and Dictionary Learning

Sparse representation problem

Terms: Sparse Representation (稀疏表現)/Sparse Coding (稀疏編碼)

SR problem: *Given a signal vector* $\boldsymbol{x} \in \mathbb{R}^m$ *and a dictionary matrix* $\boldsymbol{D} \in \mathbb{R}^{m \times n}$*, we seek a sparse coefficient vector* $\boldsymbol{z}^* \in \mathbb{R}^n$ *such that*

$$\boldsymbol{z}^* = \arg\min_{\boldsymbol{z}} \left(\frac{1}{2} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{z}\|_2^2 + \lambda \|\boldsymbol{z}\|_0 \right),$$

where $\lambda > 0$ *is a penalty parameter and* $\|\boldsymbol{z}\|_0$ *counts the number of nonzero components of* $\boldsymbol{z}$*.*

Remarks:

- In the matrix-vector multiplication $\boldsymbol{D}\boldsymbol{z}$, the components of $\boldsymbol{z}$ are the coefficients with respect to columns (also called *atoms*) of $\boldsymbol{D}$.
- We call $\|\boldsymbol{z}\|_0$ the ℓ^0 norm of $\boldsymbol{z}$, even though ℓ^0 is *not* really a norm, since the *homogeneity property* fails, $\|\alpha \boldsymbol{z}\|_0 \neq |\alpha| \|\boldsymbol{z}\|_0$.
- It is inefficient to compute $\|\boldsymbol{z}\|_0$ directly when n is large. In practice, we will use the ℓ^1 norm instead of the ℓ^0 norm.

Two dual ℓ^0 minimization problems

In [Sharon-Wright-Ma 2007], they studied the following two *dual* ℓ^0 minimization problems:

- **Sparse error correction (SEC):** *Given* $\mathbf{0} \neq \boldsymbol{y} \in \mathbb{R}^n$ *and* $\boldsymbol{A} \in \mathbb{R}^{n\times p}$ *with* $n > p$ *and* $\mathrm{rank}(\boldsymbol{A}) = p$*, we seek* $\boldsymbol{w}^* \in \mathbb{R}^p$ *such that*

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}\|_0. \tag{1}$$

- **Sparse signal reconstruction (SSR):** *Given* $\boldsymbol{D} \in \mathbb{R}^{m\times n}$ *with* $m < n$ *and* $\mathbf{0} \neq \boldsymbol{x} \in C(\boldsymbol{D})$ *the column space of* $\boldsymbol{D}$*, we seek* $\boldsymbol{z}^* \in \mathbb{R}^n$ *such that*

$$\boldsymbol{z}^* = \arg\min_{\boldsymbol{z}} \|\boldsymbol{z}\|_0 \quad \text{subject to} \quad \boldsymbol{x} = \boldsymbol{D}\boldsymbol{z}. \tag{2}$$

Note that (1) is a *decoding problem*, while (2) is a *sparse representation problem*. These two problems are *dual* in the sense that we can convert one problem to the other, see page 8 below.

Existence and uniqueness of solution

1. **Existence:**
   - Existence of $\boldsymbol{w}^*$: If $\exists\, \boldsymbol{w} \in \mathbb{R}^p$ s.t. $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}\|_0 = 0$, then $\boldsymbol{w}^* = \boldsymbol{w}$. Otherwise, define

     $$\mathcal{S} := \{k \in \mathbb{N} : \exists\, \boldsymbol{w} \in \mathbb{R}^p \text{ s.t. } \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}\|_0 = k\}.$$

     Then $\varnothing \neq \mathcal{S} \subseteq \mathbb{N}$. *By the well-ordering principle,* $\exists\, k_0 \in \mathcal{S}$ *the minimum of* $\mathcal{S}$. i.e., $\exists\, \boldsymbol{w}^*$ such that $\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}\|_0$.
   - Existence of $\boldsymbol{z}^*$: It can be shown in a similar way!

2. **Uniqueness:** *It will generally be true* that these two dual problems have a unique solution if
   - *$\exists\, \boldsymbol{w}_0$ such that the error $\boldsymbol{e} := \boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}_0$ is sparse enough.*
     The proof needs the "null space property" and the "restricted isometry property."
   - *$\exists\, \boldsymbol{z}_0$ sparse enough such that $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{z}_0$.*
     e.g., if any set of $2T$ columns of $\boldsymbol{D}$ are linearly independent, then any $\boldsymbol{z}_0 \in \mathbb{R}^n$ with $\|\boldsymbol{z}_0\|_0 \leq T$ such that $\boldsymbol{D}\boldsymbol{z}_0 = \boldsymbol{x}$ is the unique solution to SSR problem (2).

Why we require matrix A full rank p in the SEC problem?

Note that $\boldsymbol{A}$ is of size $n \times p$ and $n > p$.

Suppose that $\boldsymbol{A}$ is not full rank p. Then $\text{rank}(\boldsymbol{A}) < p$.

Since $\dim N(\boldsymbol{A}) + \text{rank}(\boldsymbol{A}) = p$, we have $\dim N(\boldsymbol{A}) > 0$.

Thus, nullspace $N(\boldsymbol{A}) \neq \{\mathbf{0}\}$ and $\exists\, \widetilde{\boldsymbol{w}} \neq \mathbf{0}$ such that $\boldsymbol{A}\widetilde{\boldsymbol{w}} = \mathbf{0}$.

If $\boldsymbol{w}^*$ is a solution of the SEC problem, then

$$\|\boldsymbol{y} - \boldsymbol{A}(\boldsymbol{w}^* + \widetilde{\boldsymbol{w}})\|_0 = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}^*\|_0.$$

Hence, $\boldsymbol{w}^* + \widetilde{\boldsymbol{w}}$ is also a solution of the SEC problem.

Therefore, in order to ensure the uniqueness, we require $\boldsymbol{A}$ full rank p.

How to convert problem (2) to problem (1)?

- *The decoding problem (1) can be converted to the sparse representation problem (2).* [Candès *et al.* 2005, IEEE Symposium on FOCS]
- **Converting (2) to (1):** *Let $p = n - \mathrm{rank}(\boldsymbol{D}) > 0$ and $\boldsymbol{A}$ be a full-rank $n \times p$ matrix whose columns span the nullspace of $\boldsymbol{D}$, i.e., $\boldsymbol{D}\boldsymbol{A} = \boldsymbol{0}$. Find any $\boldsymbol{y} \in \mathbb{R}^n$ so that $\boldsymbol{D}\boldsymbol{y} = \boldsymbol{x}$ and define $f(\boldsymbol{w}) = \boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}$. Then*

$$\underbrace{\underset{\boldsymbol{z}\ \&\ \boldsymbol{D}\boldsymbol{z}=\boldsymbol{x}}{\arg\min} \|\boldsymbol{z}\|_0}_{\boldsymbol{z}^*} = \underbrace{f\left(\underset{\boldsymbol{w}}{\arg\min} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}\|_0\right)}_{f(\boldsymbol{w}^*)}. \qquad (3)$$

Proof: First, note that for all $\boldsymbol{w} \in \mathbb{R}^p$, we have

$$\boldsymbol{D}f(\boldsymbol{w}) = \boldsymbol{D}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}) = \boldsymbol{D}\boldsymbol{y} - \boldsymbol{D}\boldsymbol{A}\boldsymbol{w} = \boldsymbol{D}\boldsymbol{y} = \boldsymbol{x} \implies \|f(\boldsymbol{w})\|_0 \geq \|\boldsymbol{z}^*\|_0.$$

Claim: $\exists\, \widetilde{\boldsymbol{w}} \in \mathbb{R}^p$ such that $f(\widetilde{\boldsymbol{w}}) = \boldsymbol{y} - \boldsymbol{A}\widetilde{\boldsymbol{w}} = \boldsymbol{z}^*$.

$\because\ \boldsymbol{D}\boldsymbol{z}^* = \boldsymbol{x}$ and $\boldsymbol{D}(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}) = \boldsymbol{x},\ \forall \boldsymbol{w} \implies \boldsymbol{D}(-\boldsymbol{z}^* + \boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}) = \boldsymbol{0}$

$\therefore\ \exists\, \bar{\boldsymbol{w}}$ such that $\boldsymbol{A}\bar{\boldsymbol{w}} = -\boldsymbol{z}^* + \boldsymbol{y} - \boldsymbol{A}\boldsymbol{w} \implies \boldsymbol{z}^* = \boldsymbol{y} - \boldsymbol{A}(\boldsymbol{w} + \bar{\boldsymbol{w}}) := f(\widetilde{\boldsymbol{w}})$

Claim: $\widetilde{\boldsymbol{w}} = \boldsymbol{w}^* := \arg\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}\|_0$, and then $f(\boldsymbol{w}^*) = \boldsymbol{z}^*$.

$\because\ \|f(\boldsymbol{w}^*)\|_0 \leq \|f(\widetilde{\boldsymbol{w}})\|_0 = \|\boldsymbol{z}^*\|_0 \leq \|f(\boldsymbol{w}^*)\|_0 \implies \|f(\boldsymbol{w}^*)\|_0 = \|f(\widetilde{\boldsymbol{w}})\|_0$

By the uniqueness of $\boldsymbol{w}^*$, we obtain $\widetilde{\boldsymbol{w}} = \boldsymbol{w}^*$ and then $f(\boldsymbol{w}^*) = \boldsymbol{z}^*$.

The ℓ^1-ℓ^0 equivalence problem

- In general, the ℓ^0 minimizations (1) and (2) are *NP-hard problems:*

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}\|_0 , \tag{1}$$

$$\boldsymbol{z}^* = \arg\min_{\boldsymbol{z}} \|\boldsymbol{z}\|_0 \quad \text{subject to} \quad \boldsymbol{x} = \boldsymbol{D}\boldsymbol{z}. \tag{2}$$

- The equivalence between ℓ^0 and ℓ^1 minimizations is conditional.

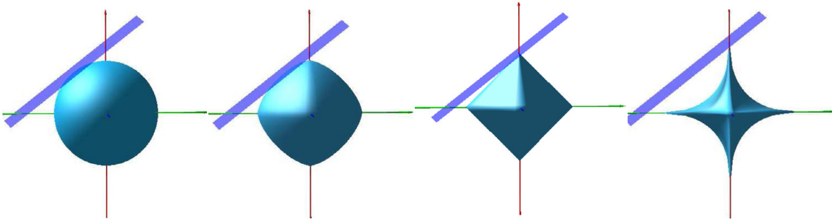
  David L. Donoho, *For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution,* CPAM, 59 (2006), pp. 797-829.

  If the error $\boldsymbol{e} := \boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}^*$ and the solution $\boldsymbol{z}^*$ is sufficiently sparse, then the solutions to (1) and (2) are the same as (4) and (5), respectively.

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{w}\|_1 , \tag{4}$$

$$\boldsymbol{z}^* = \arg\min_{\boldsymbol{z}} \|\boldsymbol{z}\|_1 \quad \text{subject to} \quad \boldsymbol{x} = \boldsymbol{D}\boldsymbol{z}. \tag{5}$$

3-D ball in different ℓ^r norms and the constraint $Dz = x$



ℓ^2-norm $\to$ $\ell^{1.5}$-norm $\to$ ℓ^1-norm $\to$ $\ell^{0.5}$-norm

3-D ball in the different ℓ^r norms for $r = 2,\ 1.5,\ 1,\ 0.5$

$$z^* = \arg\min_{z} \|z\|_1 \quad \text{subject to} \quad \underbrace{x}_{\text{given}} = D \underbrace{z}_{\text{many}} \tag{5}$$

The sparse representation problem

- We have introduced some ideas about the ℓ^1-ℓ^0 equivalence. In what follows, we don't consider the original SR problem. We consider the following ℓ^1 minimization problem instead:

  SR problem: *Given a signal vector $\boldsymbol{x} \in \mathbb{R}^m$ and a dictionary matrix $\boldsymbol{D} \in \mathbb{R}^{m\times n}$, we seek a coefficient vector $\boldsymbol{z}^* \in \mathbb{R}^n$ such that*

  $$\boldsymbol{z}^* = \arg\min_{\boldsymbol{z}} \left(\frac{1}{2} \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{z}\|_2^2 + \lambda \|\boldsymbol{z}\|_1 \right), \qquad \lambda > 0. \qquad (\star)$$

  The existence (and uniqueness) of solution of the SR problem $(\star)$ can be ensured because matrix $\boldsymbol{D}^\top \boldsymbol{D}$ is symmetric (+ *positive definite*) and the second term $\lambda\|\cdot\|_1$ is a *convex function*.

- Problem $(\star)$ is also a regression analysis method in statistics and machine learning. It is the so-called *least absolute shrinkage and selection operator (LASSO).*

  R. J. Tibshirani, The lasso problem and uniqueness, *Electronic Journal of Statistics*, 7 (2013), pp. 1456-1490 $\oplus$ A. Ali, 13 (2019), pp. 2307-2347.

Alternating direction method of multipliers (ADMM)

We will use the *"Alternating Direction Method of Multipliers"* to solve the above ℓ^1-norm SR problem.

- ADMM is an iterative scheme for solving the following equality constrained convex optimization problems:

$$\min_{\boldsymbol{z}} f(\boldsymbol{z}) \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}.$$

- ADMM consists of three steps:
  1. *adding an auxiliary variable* $\boldsymbol{y}$ *and a dual variable (multipliers)* $\boldsymbol{v}$ *and then scaled as* $\boldsymbol{u}$
  2. *separating the new cost function into a sum of* $f(\boldsymbol{z})$ *and* $g(\boldsymbol{y})$
  3. *using an iterative method to solve the problem*

- Then the optimization problem can be re-posed as

$$\min_{\boldsymbol{z},\,\boldsymbol{y}} \big(f(\boldsymbol{z}) + g(\boldsymbol{y})\big) \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}\boldsymbol{y} = \boldsymbol{c}.$$

Derivation of the ADMM: augmented Lagrangian

First, we formulate the *augmented Lagrangian*

$$L_\rho(\boldsymbol{z},\boldsymbol{y},\boldsymbol{v}) := f(\boldsymbol{z}) + g(\boldsymbol{y}) + \underbrace{\boldsymbol{v}^\top}_{\textit{multipliers}} (\boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}\boldsymbol{y} - \boldsymbol{c}) + \underbrace{\frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}\boldsymbol{y} - \boldsymbol{c}\|_2^2}_{\textit{penalty term}},$$

where $\rho > 0$ is the penalty parameter. Then the iterative scheme of the *augmented Lagrangian method (ALM)* is given by

$$\begin{aligned}
(\boldsymbol{z}^{(i+1)}, \boldsymbol{y}^{(i+1)}) &= \arg\min_{\boldsymbol{z}} L_\rho(\boldsymbol{z},\boldsymbol{y},\boldsymbol{v}^{(i)}), \\
\boldsymbol{v}^{(i+1)} &= \boldsymbol{v}^{(i)} + \rho\left(\boldsymbol{A}\boldsymbol{z}^{(i+1)} + \boldsymbol{B}\boldsymbol{y}^{(i+1)} - \boldsymbol{c}\right).
\end{aligned}$$

In ADMM, $\boldsymbol{z}$ and $\boldsymbol{y}$ are updated in an alternating or sequential fashion, which accounts for the term *alternating direction*.

$$\begin{aligned}
\boldsymbol{z}^{(i+1)} &= \arg\min_{\boldsymbol{z}} L_\rho(\boldsymbol{z},\boldsymbol{y}^{(i)},\boldsymbol{v}^{(i)}), \\
\boldsymbol{y}^{(i+1)} &= \arg\min_{\boldsymbol{y}} L_\rho(\boldsymbol{z}^{(i+1)},\boldsymbol{y},\boldsymbol{v}^{(i)}), \\
\boldsymbol{v}^{(i+1)} &= \boldsymbol{v}^{(i)} + \rho(\boldsymbol{A}\boldsymbol{z}^{(i+1)} + \boldsymbol{B}\boldsymbol{y}^{(i+1)} - \boldsymbol{c}).
\end{aligned}$$

Scaled form of the augmented Lagrangian

The ADMM can be written in a slightly different form, which is often more convenient, by combining the linear and quadratic terms in the augmented Lagrangian and scaling the dual variable (multipliers) $\boldsymbol{v}$.

Define the residual $\boldsymbol{r} := \boldsymbol{Az} + \boldsymbol{By} - \boldsymbol{c}$. Then

$$
\begin{aligned}
&\boldsymbol{v}^\top(\boldsymbol{Az} + \boldsymbol{By} - \boldsymbol{c}) + \frac{\rho}{2}\|\boldsymbol{Az} + \boldsymbol{By} - \boldsymbol{c}\|_2^2 \\
&\qquad = \boldsymbol{v}^\top \boldsymbol{r} + \frac{\rho}{2}\|\boldsymbol{r}\|_2^2 = \frac{\rho}{2}\|\boldsymbol{r} + \frac{1}{\rho}\boldsymbol{v}\|_2^2 - \frac{1}{2\rho}\|\boldsymbol{v}\|_2^2.
\end{aligned}
$$

Set $\boldsymbol{u} = \frac{1}{\rho}\boldsymbol{v}$. Then $L_\rho(\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{v}) = L_\rho(\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{u})$, and

$$
L_\rho(\boldsymbol{z}, \boldsymbol{y}, \boldsymbol{u}) = f(\boldsymbol{z}) + g(\boldsymbol{y}) + \frac{\rho}{2}\|\boldsymbol{Az} + \boldsymbol{By} - \boldsymbol{c} + \boldsymbol{u}\|_2^2 - \frac{\rho}{2}\|\boldsymbol{u}\|_2^2.
$$

ADMM: scaled form

The ADMM in the scaled form is given by

$$
\begin{aligned}
\boldsymbol{z}^{(i+1)} &= \arg\min_{\boldsymbol{z}} \Big(f(\boldsymbol{z}) + g(\boldsymbol{y}^{(i)}) + \frac{\rho}{2} \|\boldsymbol{A}\boldsymbol{z} + \boldsymbol{B}\boldsymbol{y}^{(i)} - \boldsymbol{c} + \boldsymbol{u}^{(i)}\|_2^2 - \frac{\rho}{2} \|\boldsymbol{u}^{(i)}\|_2^2 \Big), \\
\boldsymbol{y}^{(i+1)} &= \arg\min_{\boldsymbol{y}} \Big(f(\boldsymbol{z}^{(i+1)}) + g(\boldsymbol{y}) + \frac{\rho}{2} \|\boldsymbol{A}\boldsymbol{z}^{(i+1)} + \boldsymbol{B}\boldsymbol{y} - \boldsymbol{c} + \boldsymbol{u}^{(i)}\|_2^2 - \frac{\rho}{2} \|\boldsymbol{u}^{(i)}\|_2^2 \Big) \\
\boldsymbol{u}^{(i+1)} &= \boldsymbol{u}^{(i)} + \boldsymbol{A}\boldsymbol{z}^{(i+1)} + \boldsymbol{B}\boldsymbol{y}^{(i+1)} - \boldsymbol{c},
\end{aligned}
$$

where $\rho > 0$ is the *penalty parameter* which is related to the convergent rate of the iterations.

Note that the terms in blue can be omitted in practical computations!

Reference: S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, Distributed optimization and statistical learning via the ADMM, *Foundations and Trends in Machine Learning*, 3 (2010), pp. 1-122.

ADMM for the ℓ^1-norm SR problem

- For the ℓ^1-norm SR problem,

$$z^* = \arg\min_{z} \left(\frac{1}{2} \|x - Dz\|_2^2 + \lambda \|z\|_1 \right), \qquad \lambda > 0, \qquad (\star)$$

we set

$$f(z) := \frac{1}{2} \|x - Dz\|_2^2,\ g(y) := \lambda \|y\|_1,\ Az + By = c \Leftrightarrow z - y = \mathbf{0}.$$

- The ADMM for the ℓ^1-norm SR problem is given by

$$\begin{aligned}
z^{(i+1)} &= \arg\min_{z} \left(\frac{1}{2} \|x - Dz\|_2^2 + \frac{\rho}{2} \|z - y^{(i)} + u^{(i)}\|_2^2 \right), && (6_1)\\
y^{(i+1)} &= \arg\min_{y} \left(\lambda \|y\|_1 + \frac{\rho}{2} \|z^{(i+1)} - y + u^{(i)}\|_2^2 \right), && (6_2)\\
u^{(i+1)} &= u^{(i)} + z^{(i+1)} - y^{(i+1)}, && (6_3)
\end{aligned}$$

where $\rho > 0$ is *penalty parameter* related to the convergent rate of the iterations.

Solving minimization problem (6_1)

Define

$$F_1(\boldsymbol{z}) := \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{z}\|_2^2 + \frac{\rho}{2}\|\boldsymbol{z} - \boldsymbol{y}^{(i)} + \boldsymbol{u}^{(i)}\|_2^2.$$

Then F_1 is a quadratic function in variables $z_1, z_2, \cdots, z_n$ and $F_1(\boldsymbol{z}) \geq 0 \; \forall \; \boldsymbol{z} \in \mathbb{R}^n$. To solve "$\min\limits_{\boldsymbol{z}} F_1(\boldsymbol{z})$", first we compute

$$\begin{aligned} \nabla F_1(\boldsymbol{z}) &= -\boldsymbol{D}^\top(\boldsymbol{x} - \boldsymbol{D}\boldsymbol{z}) + \rho \boldsymbol{I}(\boldsymbol{z} - \boldsymbol{y}^{(i)} + \boldsymbol{u}^{(i)}) \\ &= (\boldsymbol{D}^\top\boldsymbol{D} + \rho\boldsymbol{I})\boldsymbol{z} - \left(\boldsymbol{D}^\top\boldsymbol{x} + \rho(\boldsymbol{y}^{(i)} - \boldsymbol{u}^{(i)})\right). \end{aligned}$$

Letting $\nabla F_1(\boldsymbol{z}) = \boldsymbol{0}$, we have

$$(\boldsymbol{D}^\top\boldsymbol{D} + \rho\boldsymbol{I})\boldsymbol{z} = \left(\boldsymbol{D}^\top\boldsymbol{x} + \rho(\boldsymbol{y}^{(i)} - \boldsymbol{u}^{(i)})\right).$$

Therefore, we obtain the solution

$$\boldsymbol{z}^{(i+1)} = (\boldsymbol{D}^\top\boldsymbol{D} + \rho\boldsymbol{I})^{-1}\left(\boldsymbol{D}^\top\boldsymbol{x} + \rho(\boldsymbol{y}^{(i)} - \boldsymbol{u}^{(i)})\right).$$

Solving minimization problem (6_2)

Using the *soft-thresholding function* $\mathcal{S}_{\lambda/\rho}$, the solution of problem (6_2) has the closed form (see next few pages):

$$\boldsymbol{y}^{(i+1)} = \mathcal{S}_{\lambda/\rho}(\boldsymbol{z}^{(i+1)} + \boldsymbol{u}^{(i)}),$$

where

$$\mathcal{S}_{\lambda/\rho}(\boldsymbol{v}) = \operatorname{sign}(\boldsymbol{v}) \odot \max(\boldsymbol{0}, |\boldsymbol{v}| - \lambda/\rho),$$

and $\operatorname{sign}(\cdot)$, $\max(\cdot,\cdot)$, and $|\cdot|$ are all applied to the input vector $\boldsymbol{v}$ component-wisely, and $\odot$ is the Hadamard product.

Finally, the iterative scheme can be posed as follows:

$$\begin{aligned}
\boldsymbol{z}^{(i+1)} &= (\boldsymbol{D}^\top \boldsymbol{D} + \rho \boldsymbol{I})^{-1}(\boldsymbol{D}^\top \boldsymbol{x} + \rho(\boldsymbol{y}^{(i)} - \boldsymbol{u}^{(i)})), && (7_1)\\
\boldsymbol{y}^{(i+1)} &= \mathcal{S}_{\lambda/\rho}(\boldsymbol{z}^{(i+1)} + \boldsymbol{u}^{(i)}), && (7_2)\\
\boldsymbol{u}^{(i+1)} &= \boldsymbol{u}^{(i)} + \boldsymbol{z}^{(i+1)} - \boldsymbol{y}^{(i+1)}. && (7_3)
\end{aligned}$$

Details of the solution of problem (6_2)

Recall the problem (6_2),

$$\boldsymbol{y}^{(i+1)} = \arg\min_{\boldsymbol{y}} \left(\lambda \|\boldsymbol{y}\|_1 + \frac{\rho}{2} \|\boldsymbol{z}^{(i+1)} - \boldsymbol{y} + \boldsymbol{u}^{(i)}\|_2^2 \right). \qquad (6_2)$$

Let $\boldsymbol{v} := \boldsymbol{z}^{(i+1)} + \boldsymbol{u}^{(i)} \in \mathbb{R}^n$. Then we have

$$\boldsymbol{y}^{(i+1)} = \arg\min_{\boldsymbol{y}} \left(\lambda \|\boldsymbol{y}\|_1 + \frac{\rho}{2} \|\boldsymbol{v} - \boldsymbol{y}\|_2^2 \right).$$

Define a real-valued function $F_2(\boldsymbol{y})$ as follows:

$$\begin{aligned} F_2(\boldsymbol{y}) &= \lambda \|\boldsymbol{y}\|_1 + \frac{\rho}{2} \|\boldsymbol{v} - \boldsymbol{y}\|_2^2 \\ &= \left(\lambda |y_1| + \frac{\rho}{2} (v_1 - y_1)^2 \right) + \cdots + \left(\lambda |y_n| + \frac{\rho}{2} (v_n - y_n)^2 \right) \\ &:= f_1(y_1) + \cdots + f_n(y_n), \end{aligned}$$

where we define

$$f_j(y) := \lambda |y| + \frac{\rho}{2} (v_j - y)^2 \quad \forall\, j = 1, 2, \cdots, n.$$

Analysis of functions f_j

For simplicity of the presentation, we consider the function

$$f(y) = \lambda|y| + \frac{\rho}{2}(v-y)^2.$$

Computing the derivative of $f(y)$ for $y \neq 0$, we have

$$f'(y) = \begin{cases} \lambda - \rho(v-y) & y > 0, \\ -\lambda - \rho(v-y) & y < 0, \end{cases}$$

Let $f'(y) = 0$. Then we have the critical number $c \neq 0$,

$$c = \begin{cases} v - \frac{\lambda}{\rho} & \text{if } v > \frac{\lambda}{\rho}, \quad (y > 0) \\ v + \frac{\lambda}{\rho} & \text{if } v < -\frac{\lambda}{\rho} \quad (y < 0). \end{cases}$$

In order to find the minimum of f, we consider the following three cases:

$$v > \frac{\lambda}{\rho}, \quad v < -\frac{\lambda}{\rho}, \quad |v| \leq \frac{\lambda}{\rho}.$$

Case 1: $v > \frac{\lambda}{\rho}$

In this case, the critical number is $c = v - \frac{\lambda}{\rho} > 0$, and

$$\begin{aligned} f(c) = f(v - \frac{\lambda}{\rho}) &= \lambda(v - \frac{\lambda}{\rho}) + \frac{\rho}{2}(v - (v - \frac{\lambda}{\rho}))^2 \\ &= \frac{\rho}{2}\left(v^2 - (v - \frac{\lambda}{\rho})^2\right) < \frac{\rho}{2}v^2 = f(0). \end{aligned}$$

For $y \geq 0$, since f is a quadratic polynomial in y with positive leading coefficient, we can conclude that $f(c) \leq f(y)$ for all $y \geq 0$.

For $y < 0$, $f(y)$ is monotone decreasing since

$$\begin{aligned} f'(y) &= -\lambda - \rho(v - y) = -\lambda - \rho v + \rho y \\ &< -\lambda - \lambda + \rho y = -2\lambda + \rho y < 0, \end{aligned}$$

which implies $f(y) > f(0)$ for all $y < 0$.

Therefore, f has a minimum at $c = v - \frac{\lambda}{\rho} > 0$.

Case 2: $v < -\frac{\lambda}{\rho}$

In this case, the critical number is $c = v + \frac{\lambda}{\rho} < 0$, and

$$\begin{aligned} f(c) = f(v + \frac{\lambda}{\rho}) &= -\lambda(v + \frac{\lambda}{\rho}) + \frac{\rho}{2}\left(v - (v + \frac{\lambda}{\rho})\right)^2 \\ &= \frac{\rho}{2}\left(v^2 - (v + \frac{\lambda}{\rho})^2\right) < \frac{\rho}{2}v^2 = f(0). \end{aligned}$$

For $y \leq 0$, since f is a quadratic polynomial in y with positive leading coefficient, we can conclude that $f(c) \leq f(y)$ for all $y \leq 0$.

For $y > 0$, $f(y)$ is monotone increasing since

$$\begin{aligned} f'(y) &= \lambda - \rho(v - y) = \lambda - \rho v + \rho y \\ &> \lambda + \lambda + \rho y = 2\lambda + \rho y > 0, \end{aligned}$$

which implies $f(y) > f(0)$ for all $y > 0$.

Therefore, f has a minimum at $c = v + \frac{\lambda}{\rho} > 0$.

Case 3: $|v| \le \frac{\lambda}{\rho}$

In this case, the only critical number is $y = 0$. For $y > 0$, $f(y)$ is monotone increasing since

$$\begin{aligned} f'(y) &= \lambda - \rho(v - y) = \lambda - \rho v + \rho y \\ &\ge \lambda - \lambda + \rho y = \rho y > 0, \end{aligned}$$

which implies $f(y) > f(0)$ for all $y > 0$.

For $y < 0$, $f(y)$ is monotone decreasing since

$$\begin{aligned} f'(y) &= -\lambda - \rho(v - y) = -\lambda - \rho v + \rho y \\ &\le -\lambda + \lambda + \rho y = \rho y < 0, \end{aligned}$$

which implies $f(y) > f(0)$ for all $y < 0$.

Therefore, f has a minimum at 0.

Solution of problem (6_2)

By the above discussions, we have

$$\arg\min_y f(y) = \begin{cases} v + \frac{\lambda}{\rho}, & \text{if } v < -\frac{\lambda}{\rho}, \qquad \text{(case 2)} \\ 0, & \text{if } |v| \le \frac{\lambda}{\rho}, \qquad \text{(case 3)} \\ v - \frac{\lambda}{\rho}, & \text{if } v > \frac{\lambda}{\rho}. \qquad \text{(case 1)} \end{cases}$$

In other words, we have

$$\arg\min_y f(y) = \mathcal{S}_{\lambda/\rho}(v) = \mathrm{sign}(v)\max(0, |v| - \lambda/\rho).$$

Therefore,

$$\boldsymbol{y}^{(i+1)} = \arg\min_{\boldsymbol{y}} F_2(\boldsymbol{y}) = \mathcal{S}_{\lambda/\rho}(\boldsymbol{v}) = \mathcal{S}_{\lambda/\rho}(\boldsymbol{z}^{(i+1)} + \boldsymbol{u}^{(i)}).$$

where the soft-thresholding,

$$\mathcal{S}_{\lambda/\rho}(\boldsymbol{v}) := \mathrm{sign}(\boldsymbol{v}) \odot \max(\boldsymbol{0},\ |\boldsymbol{v}| - \lambda/\rho),$$

and $\mathrm{sign}(\cdot)$, $\max(\cdot,\cdot)$, and $|\cdot|$ are all applied to the input vector $\boldsymbol{v}$ component-wisely, and $\odot$ is the Hadamard product.

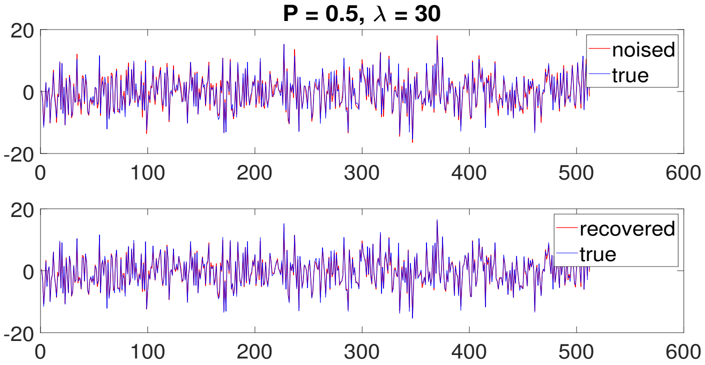
Application to signal denoising

- First, we construct a random dictionary matrix $\boldsymbol{D} \in \mathbb{R}^{512\times 2048}$ and a random sparse vector $\boldsymbol{z} \in \mathbb{R}^{2048}$ with $\|\boldsymbol{z}\|_0 = 32$. We then have the true signal $\boldsymbol{x} := \boldsymbol{D}\boldsymbol{z}$.
- Define the noise signal $\boldsymbol{x}_n := \boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{n} \in \mathbb{R}^{512}$ is a random white Gaussian noise with noised powers $P = 0.5, 1, 5$. We then consider $\lambda = 5, 10, 20, 30$ for the minimization problem.
- *Peak signal-to-noise ratio (PSNR):* We define the mean squared error (MSE) and then the PSNR as follows:

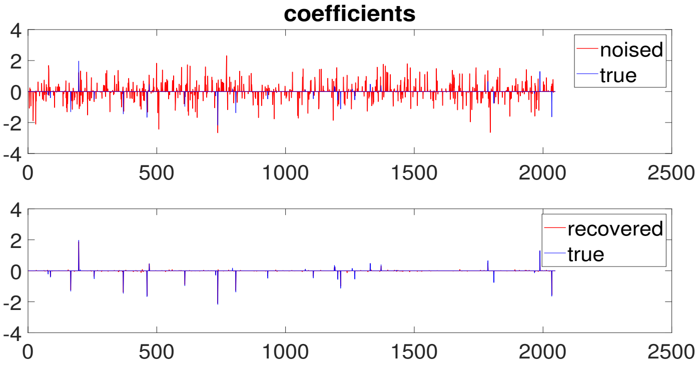
$$\begin{aligned} MSE &:= \frac{1}{512}\sum_{i=1}^{512}\left(\text{true}(i) - \text{approx}(i)\right)^2, \\ PSNR &:= 10 \times \log_{10}\left(\frac{\max^2}{MSE}\right), \end{aligned}$$

  where "max" is the maximum amplitude of the true signal.
- Source of matlab code: `http://brendt.wohlberg.net/software/SPORCO/`

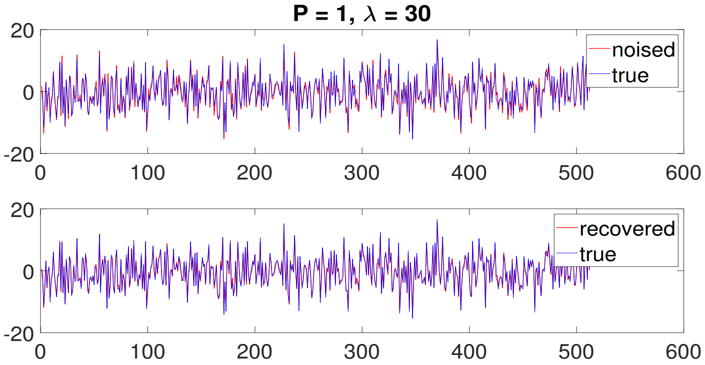
Numerical results for $P = 0.5$ and $\lambda = 30$



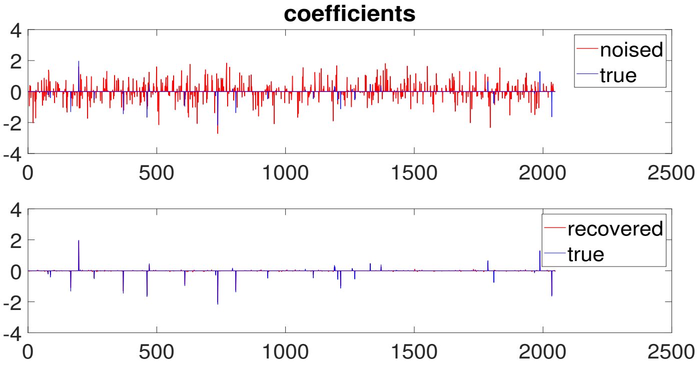
Coefficients for $P = 0.5$ and $\lambda = 30$



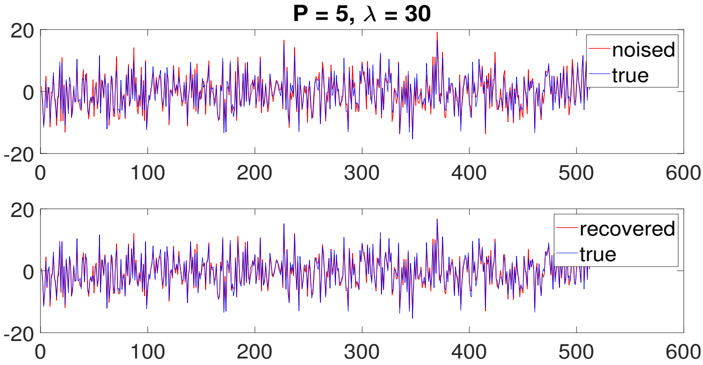
Numerical results for $P = 1$ and $\lambda = 30$



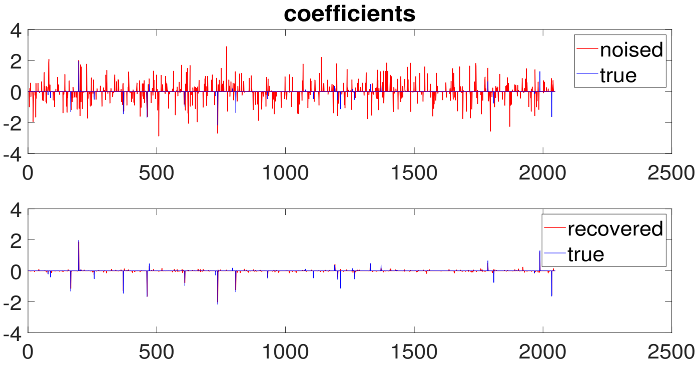
Coefficients for $P = 1$ and $\lambda = 30$



Numerical result for $P = 5$ and $\lambda = 30$



Coefficients for $P = 5$ and $\lambda = 30$



PSNR values and iteration numbers

In general, the higher the value of PSNR the better the quality of the recovered signals.

PSNR values

P	0.5	0.5	1	1	5	5
λ	noised	rcvered	noised	rcvered	noised	rcvered
5	29.51	30.36	29.71	30.41	25.57	26.11
10	29.51	31.16	29.71	31.10	25.57	26.63
20	29.51	32.55	29.71	32.23	25.57	27.62
30	29.51	**33.45**	29.71	**32.77**	25.57	**28.50**

Iteration numbers of ADMM

$\lambda \backslash P$	0.5	1	5
5	550	664	569
10	301	303	320
20	172	169	186
30	**129**	**130**	**154**

Sparse dictionary learning problem

In the SR problem, the solution of interest $\boldsymbol{z}^*$ *is the coefficient vector of a linear combination of over-complete basis elements (columns) from a given dictionary* $\mathbf{D}$ *under some sparsity constraint. Therefore, it is typically accompanied by a dictionary learning mechanism.*

We are going to study a more general problem. The dictionary $\boldsymbol{D}$ is unknown and needed to be sought together with the sparse solution.

SDL problem: *Let* $\{\boldsymbol{x}_i\}_{i=1}^N \subset \mathbb{R}^m$ *be a given dataset of signals. We seek a dictionary matrix* $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n] \in \mathbb{R}^{m \times n}$ *together with the sparse coefficient vectors* $\{\boldsymbol{z}_i\}_{i=1}^N \subset \mathbb{R}^n$ *that solve the minimization problem:*

$$\min_{\boldsymbol{D}, \{\boldsymbol{z}_i\}} \left(\frac{1}{2} \sum_{i=1}^N \|\boldsymbol{x}_i - \boldsymbol{D}\boldsymbol{z}_i\|_2^2 + \lambda \sum_{i=1}^N \|\boldsymbol{z}_i\|_1 \right)$$

$$\text{subject to } \|\boldsymbol{d}_k\|_2 \le 1, \ \forall\, 1 \le k \le n,$$

where $\lambda > 0$ *is a penalty parameter.*

Problem formulation in a more compact form

To simplify the formulation of the SDL problem, we define

$$
\begin{aligned}
\boldsymbol{X} &= [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{m\times N}, \\
\boldsymbol{Z} &= [\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_N] \in \mathbb{R}^{n\times N}.
\end{aligned}
$$

Then the SDL problem can be posed as follows: *Given a training data matrix $\boldsymbol{X}$, find a dictionary matrix $\boldsymbol{D}$ and a coefficient matrix $\boldsymbol{Z}$ such that*

$$
\min_{\boldsymbol{D},\,\boldsymbol{Z}} \left(\frac{1}{2} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{Z}\|_F^2 + \lambda \|\boldsymbol{Z}\|_{1,1} \right) \qquad (\star\star)
$$
$$
\text{subject to } \|\boldsymbol{d}_k\|_2 \le 1, \ \forall\, 1 \le k \le n,
$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $\|\boldsymbol{Z}\|_{1,1}$ is the $L_{1,1}$-norm which is defined as

$$
\|\boldsymbol{Z}\|_{1,1} := \sum_{i=1}^{N} \|\boldsymbol{z}_i\|_1.
$$

An iterative approach for solving the SDL problem

In the SDL problem $(\star\star)$, we have two unknown matrices $\boldsymbol{D}$ and $\boldsymbol{Z}$. We will use a simple iterative approach together with the ADMM to solve $(\star\star)$, though it is more complicated.

Given an initial guess $\boldsymbol{D}_{(0)}$, for $j = 0, 1, \cdots$, we solve the following two sub-problems alternatingly:

$$\boldsymbol{Z}_{(j)} = \arg\min_{\boldsymbol{Z}} \left(\frac{1}{2} \|\boldsymbol{X} - \boldsymbol{D}_{(j)}\boldsymbol{Z}\|_F^2 + \lambda \|\boldsymbol{Z}\|_{1,1} \right), \tag{8}$$

$$\begin{aligned} \boldsymbol{D}_{(j+1)} = \arg\min_{\boldsymbol{D}} & \left(\frac{1}{2} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{Z}_{(j)}\|_F^2 + \lambda \|\boldsymbol{Z}_{(j)}\|_{1,1} \right) \\ & \text{subject to } \|\boldsymbol{d}_k\|_2 \le 1, \ \forall\, 1 \le k \le n. \end{aligned} \tag{9}$$

We iterate (8) and (9) until convergence is achieved. As we have introduced previously, problems (8) and (9) will be solved by ADMM.

ADMM for solving problem (8)

- Adding an *auxiliary variable* $\boldsymbol{Y}$ and a *dual variable* $\boldsymbol{U}$, we define

$$f(\boldsymbol{Z}) := \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{D}_{(j)}\boldsymbol{Z}\|_F^2, \quad g(\boldsymbol{Y}) := \lambda\|\boldsymbol{Y}\|_{1,1}, \quad \boldsymbol{Z} = \boldsymbol{Y}.$$

- Then the ADMM for solving (8) is given by

$$\begin{aligned}
\boldsymbol{Z}^{(i+1)} &= \arg\min_{\boldsymbol{Z}} \left(\frac{1}{2}\|\boldsymbol{X} - \boldsymbol{D}_{(j)}\boldsymbol{Z}\|_F^2 + \frac{\rho}{2}\|\boldsymbol{Z} - \boldsymbol{Y}^{(i)} + \boldsymbol{U}^{(i)}\|_F^2 \right), \quad (8_1) \\
\boldsymbol{Y}^{(i+1)} &= \arg\min_{\boldsymbol{Y}} \left(\lambda\|\boldsymbol{Y}\|_{1,1} + \frac{\rho}{2}\|\boldsymbol{Z}^{(i+1)} - \boldsymbol{Y} + \boldsymbol{U}^{(i)}\|_F^2 \right), \quad (8_2) \\
\boldsymbol{U}^{(i+1)} &= \boldsymbol{U}^{(i)} + \boldsymbol{Z}^{(i+1)} - \boldsymbol{Y}^{(i+1)}. \quad (8_3).
\end{aligned}$$

- Similar to the SR problem, we will use the same methods to solve the sub-problems (8_1) and (8_2).

Solving minimization problem (8_1)

Define

$$F_1(\boldsymbol{Z}) := \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{D}_{(j)}\boldsymbol{Z}\|_F^2 + \frac{\rho}{2}\|\boldsymbol{Z} - \boldsymbol{Y}^{(i)} + \boldsymbol{U}^{(i)}\|_F^2.$$

To solve "$\min_{\boldsymbol{Z}} F_1(\boldsymbol{Z})$", first we compute

$$\begin{aligned} \nabla F_1(\boldsymbol{Z}) &= -\boldsymbol{D}_{(j)}^\top(\boldsymbol{X} - \boldsymbol{D}_{(j)}\boldsymbol{Z}) + \rho\boldsymbol{I}(\boldsymbol{Z} - \boldsymbol{Y}^{(i)} + \boldsymbol{U}^{(i)}) \\ &= (\boldsymbol{D}_{(j)}^\top\boldsymbol{D}_{(j)} + \rho\boldsymbol{I})\boldsymbol{Z} - \left(\boldsymbol{D}_{(j)}^\top\boldsymbol{X} + \rho(\boldsymbol{Y}^{(i)} - \boldsymbol{U}^{(i)})\right). \end{aligned}$$

Letting $\nabla F_1(\boldsymbol{Z}) = 0$, we have

$$(\boldsymbol{D}_{(j)}^\top\boldsymbol{D}_{(j)} + \rho\boldsymbol{I})\boldsymbol{Z} = \left(\boldsymbol{D}_{(j)}^\top\boldsymbol{X} + \rho(\boldsymbol{Y}^{(i)} - \boldsymbol{U}^{(i)})\right).$$

Therefore, we obtain the solution

$$\boldsymbol{Z}^{(i+1)} = (\boldsymbol{D}_{(j)}^\top\boldsymbol{D}_{(j)} + \rho\boldsymbol{I})^{-1}\left(\boldsymbol{D}_{(j)}^\top\boldsymbol{X} + \rho(\boldsymbol{Y}^{(i)} - \boldsymbol{U}^{(i)})\right).$$

Solving minimization problem (8_2)

Using the component-wise soft-thresholding function, the solution of problem (8_2) has the closed form:

$$\boldsymbol{Y}^{(i+1)} = \mathcal{S}_{\lambda/\rho}(\boldsymbol{Z}^{(i+1)} + \boldsymbol{U}^{(i)}),$$

where

$$\mathcal{S}_{\lambda/\rho}(\boldsymbol{V}) = \mathrm{sign}(\boldsymbol{V}) \odot \max(\boldsymbol{0}, |\boldsymbol{V}| - \lambda/\rho),$$

with $\mathrm{sign}(\boldsymbol{V})$ and $|\boldsymbol{V}|$ are element-wisely applied to the matrix $\boldsymbol{V}$ and $\odot$ is the Hadamard product.

Therefore, the iterative scheme can be posed as follows:

$$\begin{aligned}
\boldsymbol{Z}^{(i+1)} &= (\boldsymbol{D}_{(j)}^{\top}\boldsymbol{D}_{(j)} + \rho \boldsymbol{I})^{-1}(\boldsymbol{D}_{(j)}^{\top}\boldsymbol{X} + \rho(\boldsymbol{Y}^{(i)} - \boldsymbol{U}^{(i)})), \qquad (10_1)\\
\boldsymbol{Y}^{(i+1)} &= \mathcal{S}_{\lambda/\rho}(\boldsymbol{Z}^{(i+1)} + \boldsymbol{U}^{(i)}), \qquad (10_2)\\
\boldsymbol{U}^{(i+1)} &= \boldsymbol{U}^{(i)} + \boldsymbol{Z}^{(i+1)} - \boldsymbol{Y}^{(i+1)}. \qquad (10_3)
\end{aligned}$$

Solving minimization problem (9)

Recall that

$$\begin{aligned} \boldsymbol{D}_{(j+1)} &= \arg\min_{\boldsymbol{D}} \left(\frac{1}{2} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{Z}_{(j)}\|_F^2 + \lambda \|\boldsymbol{Z}_{(j)}\|_{1,1} \right) \\ &\qquad \text{subject to } \|\boldsymbol{d}_k\|_2 \le 1, \ \forall\, 1 \le k \le n. \end{aligned} \tag{9}$$

Since the term $\lambda\|\boldsymbol{Z}_{(j)}\|_{1,1}$ is a fixed number when $\boldsymbol{Z}_{(j)}$ is given, problem (9) can be replaced by

$$\begin{aligned} \boldsymbol{D}_{(j+1)} &= \arg\min_{\boldsymbol{D}} \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{Z}_{(j)}\|_F^2 \\ &\qquad \text{subject to } \|\boldsymbol{d}_k\|_2 \le 1, \ \forall\, 1 \le k \le n. \end{aligned} \tag{9'}$$

Next, we introduce an *auxiliary variable* $\boldsymbol{G}$ and a *dual variable* $\boldsymbol{H}$ in ADMM for solving (9′).

ADMM for solving problem (9′)

Define

$$
\begin{aligned}
\boldsymbol{g}(\boldsymbol{G}) &:= \{[\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n] : \|\boldsymbol{d}_k\|_2 \leq 1, \ \forall\, 1 \leq k \leq n\}, \\
\boldsymbol{G} &:= \boldsymbol{D}.
\end{aligned}
$$

The ADMM for solving problem (9′) is given by

$$
\begin{aligned}
\boldsymbol{D}^{(i+1)} &= \arg\min_{\boldsymbol{D}} \left(\frac{1}{2} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{Z}_{(j)}\|_F^2 + \frac{\rho}{2} \|\boldsymbol{D} - \boldsymbol{G}^{(i)} + \boldsymbol{H}^{(i)}\|_F^2 \right), \quad (9_1) \\
\boldsymbol{G}^{(i+1)} &= \mathrm{proj}_{\boldsymbol{g}(\boldsymbol{G})} \{\boldsymbol{D}^{(i+1)}\}, \quad (9_2) \\
\boldsymbol{H}^{(i+1)} &= \boldsymbol{H}^{(i)} + \boldsymbol{D}^{(i+1)} - \boldsymbol{G}^{(i+1)}. \quad (9_3)
\end{aligned}
$$

For solving problem (9_1), we define

$$
F_2(\boldsymbol{D}) := \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{D}\boldsymbol{Z}_{(j)}\|_F^2 + \frac{\rho}{2} \|\boldsymbol{D} - \boldsymbol{G}^{(i)} + \boldsymbol{H}^{(i)}\|_F^2.
$$

Solving minimization problem (9_1)

Computing $\nabla F_2(\boldsymbol{D})$, we have

$$\begin{aligned}\nabla F_2(\boldsymbol{D}) &= (\boldsymbol{X}-\boldsymbol{D}\boldsymbol{Z}_{(j)})(-\boldsymbol{Z}_{(j)}^\top)+\rho\boldsymbol{I}_m(\boldsymbol{D}-\boldsymbol{G}^{(i)}+\boldsymbol{H}^{(i)})\\ &= \boldsymbol{D}(\rho\boldsymbol{I}_n+\boldsymbol{Z}_{(j)}\boldsymbol{Z}_{(j)}^\top)+\boldsymbol{X}\boldsymbol{Z}_{(j)}^\top-\rho(\boldsymbol{G}^{(i)}-\boldsymbol{H}^{(i)}).\end{aligned}$$

Letting $\nabla F_2(\boldsymbol{D})=\boldsymbol{0}$, we have

$$\boldsymbol{D}(\boldsymbol{Z}_{(j)}\boldsymbol{Z}_{(j)}^\top+\rho\boldsymbol{I}_n)=\boldsymbol{X}\boldsymbol{Z}_{(j)}^\top-\rho(\boldsymbol{G}^{(i)}-\boldsymbol{H}^{(i)}).$$

Therefore, we obtain the solution

$$\boldsymbol{D}^{(i+1)}=\left(\boldsymbol{X}\boldsymbol{Z}_{(j)}{}^\top-\rho(\boldsymbol{G}^{(i)}-\boldsymbol{H}^{(i)})\right)\left(\boldsymbol{Z}_{(j)}\boldsymbol{Z}_{(j)}{}^\top+\rho\boldsymbol{I}_n\right)^{-1}.$$

Finally, the ADMM for problem (9’) is given by

$$\begin{aligned}\boldsymbol{D}^{(i+1)} &= \left(\boldsymbol{X}\boldsymbol{Z}_{(j)}{}^\top-\rho(\boldsymbol{G}^{(i)}-\boldsymbol{H}^{(i)})\right)\left(\boldsymbol{Z}_{(j)}\boldsymbol{Z}_{(j)}{}^\top+\rho\boldsymbol{I}_n\right)^{-1}, \quad (11_1)\\ \boldsymbol{G}^{(i+1)} &= \mathrm{proj}_{\boldsymbol{g}(\boldsymbol{G})}\{\boldsymbol{D}^{(i+1)}\}, \quad (11_2)\\ \boldsymbol{H}^{(i+1)} &= \boldsymbol{H}^{(i)}+\boldsymbol{D}^{(i+1)}-\boldsymbol{G}^{(i+1)}. \quad (11_3)\end{aligned}$$

Convergence and stopping criterion

- In [Boyd *et al.* 2010], there are more details about convergence results of the ADMM.
- In the iterative scheme (10_1), (10_2), (10_3), we define

$$R_z = Y^{(i+1)} - Y^{(i)}, \quad S_z = U^{(i+1)} - U^{(i)}.$$

If R_z and S_z less than the tolerances ε_{R_z} and ε_{S_z}, then we say that the iteration of coefficients $Z^{(i+1)}$ converges.

In the iterative scheme (11_1), (11_2), (11_3), we define

$$R_d = G^{(i+1)} - G^{(i)}, \quad S_d = H^{(i+1)} - H^{(i)}.$$

If R_d and S_d less than the tolerances ε_{R_d} and ε_{S_d}, then we say that the iteration of dictionary $D^{(i+1)}$ converges.

References and source codes

ℓ^1-ℓ^0 equivalence problem:

[1] D. L. Donoho, For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution, *Communications on Pure and Applied Mathematics, 59 (2006), pp. 797-829.*

[2] Y. Sharon, J. Wright, and Y. Ma, Computation and relaxation of conditions for equivalence between ℓ^1 and ℓ^0 minimization, *UIUC Technical Report UILU-ENG-07-2008, 2007.*

ADMM:

[1] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, Distributed optimization and statistical learning via the ADMM, *Foundations and Trends in Machine Learning, 3 (2010), pp. 1-122.*

Sparse dictionary learning:

https://en.wikipedia.org/wiki/Sparse_dictionary_learning

Matlab codes: http://brendt.wohlberg.net/software/SPORCO/

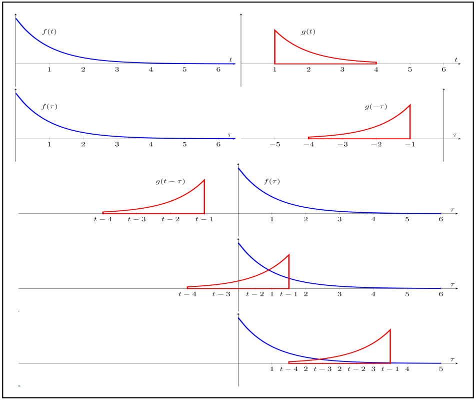
Part II

Convolutional Sparse Representation and Dictionary Learning

Convolution of two functions

Let f and g be two integrable functions with compact supports in $\mathbb{R}$. Then the convolution of f and g is defined as a function in variable t,

$$(f * g)(t) := \int_{-\infty}^{\infty} f(\tau) g(t - \tau)\, d\tau, \quad t \in \mathbb{R}.$$



Convolution of two vectors

Definition: *Let* $\boldsymbol{u} = [u_1, \cdots, u_n]^\top \in \mathbb{R}^n$ *and* $\boldsymbol{v} = [v_1, \cdots, v_m]^\top \in \mathbb{R}^m$. *The convolution of* $\boldsymbol{u}$ *and* $\boldsymbol{v}$*, denoted by* $\boldsymbol{u} * \boldsymbol{v}$*, is defined as follows:*

$$\boldsymbol{u} * \boldsymbol{v} := \begin{bmatrix} u_1 v_1 \\ u_1 v_2 + u_2 v_1 \\ u_1 v_3 + u_2 v_2 + u_3 v_1 \\ \vdots \\ u_{n-2} v_m + u_{n-1} v_{m-1} + u_n v_{m-2} \\ u_{n-1} v_m + u_n v_{m-1} \\ u_n v_m \end{bmatrix} \in \mathbb{R}^{m+n-1}.$$

More specifically, for $i = 1, 2, \cdots, (m+n-1)$*, the i-th component of* $\boldsymbol{u} * \boldsymbol{v}$ *is given by*

$$(\boldsymbol{u} * \boldsymbol{v})_i = \sum_{j=\max(1, i-m+1)}^{\min(i,n)} u_j v_{i-j+1}.$$

Remark: Convolutional operator $*$ is commutative, i.e., $\boldsymbol{u} * \boldsymbol{v} = \boldsymbol{v} * \boldsymbol{u}$.

Convolutional sparse representation (CSR) problem

CSR problem: *Given a signal $\boldsymbol{x} \in \mathbb{R}^m$ and a dictionary $\boldsymbol{D} = [\boldsymbol{d}_1, \cdots, \boldsymbol{d}_n] \in \mathbb{R}^{\ell \times n}$, we seek a sparse matrix $\boldsymbol{Z} = [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n] \in \mathbb{R}^{k \times n}$, $m = \ell + k - 1$, which solves the following minimization problem:*

$$\min_{\boldsymbol{Z}} \left(\frac{1}{2} \|\boldsymbol{x} - \sum_{j=1}^{n} \boldsymbol{d}_j * \boldsymbol{z}_j\|_2^2 + \lambda \sum_{j=1}^{n} \|\boldsymbol{z}_j\|_1 \right),$$

where $\lambda > 0$ is a penalty parameter.

Remarks:

- In SR, we use $\boldsymbol{D}\boldsymbol{z}$ to recover the signal $\boldsymbol{x}$,

$$\boldsymbol{x} \approx \boldsymbol{D}\boldsymbol{z} = \boldsymbol{d}_1 z_1 + \boldsymbol{d}_2 z_2 + \cdots + \boldsymbol{d}_n z_n = \sum_{j=1}^{n} \boldsymbol{d}_j z_j.$$

  In CSR, we use $\sum_{j=1}^{n} \boldsymbol{d}_j * \boldsymbol{z}_j$ instead,

$$\boldsymbol{x} \approx \boldsymbol{d}_1 * \boldsymbol{z}_1 + \boldsymbol{d}_2 * \boldsymbol{z}_2 + \cdots + \boldsymbol{d}_n * \boldsymbol{z}_n = \sum_{j=1}^{n} \boldsymbol{d}_j * \boldsymbol{z}_j.$$

- Convolution is a way to regulate $\boldsymbol{d}_j * \boldsymbol{z}_j$ such that $\boldsymbol{x} \approx \sum_{j=1}^{n} \boldsymbol{d}_j * \boldsymbol{z}_j$. It is more flexible than $\boldsymbol{x} \approx \sum_{j=1}^{n} \boldsymbol{d}_j z_j$, but indeed more expensive!

Toeplitz matrix

We define an $(m+n-1) \times m$ matrix $\boldsymbol{U}$ in terms of u_i, which is called a *Toeplitz matrix,* as follows:

$$\boldsymbol{U} := \begin{bmatrix} u_1 & 0 & \cdots & 0 & 0 \\ u_2 & u_1 & \ddots & 0 & 0 \\ \vdots & u_2 & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & u_1 & 0 \\ u_{n-1} & \vdots & \vdots & u_2 & u_1 \\ u_n & u_{n-1} & \vdots & \vdots & u_2 \\ 0 & u_n & \ddots & \vdots & \vdots \\ 0 & 0 & \ddots & u_{n-1} & \vdots \\ \vdots & \vdots & \ddots & u_n & u_{n-1} \\ 0 & 0 & \cdots & 0 & u_n \end{bmatrix}_{(m+n-1)\times m}$$

Then one can check that $\boldsymbol{u} * \boldsymbol{v} = \boldsymbol{U}\boldsymbol{v}$, where $\boldsymbol{u} = [u_1, \cdots, u_n]^\top \in \mathbb{R}^n$ and $\boldsymbol{v} = [v_1, \cdots, v_m]^\top \in \mathbb{R}^m$.

CSR problem using Toeplitz matrices

With the help of Toeplitz matrix, we can rewrite the CSR problem as

$$\min_{\widetilde{z}} \left(\frac{1}{2} \left\| x - \widetilde{D}\widetilde{z} \right\|_2^2 + \lambda \|\widetilde{z}\|_1 \right), \qquad (12)$$

with

$$\widetilde{z} = [z_1^\top, z_2^\top, \cdots, z_n^\top]_{nk\times 1}^\top \quad \text{and} \quad \widetilde{D} = [D_1, D_2, \cdots, D_n]_{(\ell+k-1)\times nk},$$

where D_j is a Toeplitz $(\ell + k - 1) \times k$ matrix associated with the column vector $d_j \in \mathbb{R}^\ell$, and $\ell + k - 1 = m$.

Remarks:

- We can use the same way for SR problem to solve the CSR problem (12). We can employ the ADMM, but it is too expensive since the matrix size of $\widetilde{D}$ is too large.
- *The discrete Fourier transform $\mathcal{F} : \mathbb{C}^N \to \mathbb{C}^N$ can help us to address this computational issue.*

Discrete Fourier transform (DFT) and its inverse (IDFT)

- $\widehat{\boldsymbol{x}} = \mathcal{F}(\boldsymbol{x})$: The DFT $\mathcal{F} : \mathbb{C}^N \to \mathbb{C}^N$ transforms a finite vector $\boldsymbol{x} = [x_1, x_2, \cdots, x_N]^\top$ into another vector $\widehat{\boldsymbol{x}} = [\widehat{x}_1, \widehat{x}_2, \cdots, \widehat{x}_N]^\top$, which is defined by

$$\widehat{x}_k = \sum_{n=1}^{N} x_n e^{-\frac{i2\pi}{N}(k-1)(n-1)}.$$

Then DFT is an invertible linear transformation.

- $\boldsymbol{x} = \mathcal{F}^{-1}(\widehat{\boldsymbol{x}})$: The inverse discrete Fourier transform (IDFT) $\mathcal{F}^{-1} : \mathbb{C}^N \to \mathbb{C}^N, \widehat{\boldsymbol{x}} \mapsto \boldsymbol{x}$, is given by

$$x_n = \frac{1}{N} \sum_{k=1}^{N} \widehat{x}_k e^{\frac{i2\pi}{N}(k-1)(n-1)}.$$

- Euler's formula: $e^{i\theta} = \cos\theta + i\sin\theta, \forall\, \theta \in \mathbb{R}$.

https://en.wikipedia.org/wiki/Discrete_Fourier_transform

Hadamard product

- Let $\boldsymbol{u} \in \mathbb{R}^n$ and $\boldsymbol{v} \in \mathbb{R}^m$. Then $\boldsymbol{u} * \boldsymbol{v} \in \mathbb{R}^{m+n-1}$ and

$$\mathcal{F}(\boldsymbol{u} * \boldsymbol{v}) = \mathcal{F}(\boldsymbol{u}') \odot \mathcal{F}(\boldsymbol{v}'),$$

  where $\mathcal{F}$ denotes the DFT, $\boldsymbol{u}'$ and $\boldsymbol{v}'$ are respectively *the zero padding of* $\boldsymbol{u}$ *and* $\boldsymbol{v}$ with the same size of $\boldsymbol{u} * \boldsymbol{v}$, i.e.,

$$\boldsymbol{u}' = [\boldsymbol{u}^\top, 0, \cdots, 0]^\top, \quad \boldsymbol{v}' = [\boldsymbol{v}^\top, 0, \cdots, 0]^\top \in \mathbb{R}^{m+n-1},$$

  and $\odot$ is the Hadamard product.
- The Hadamard product $\odot$ of two vectors is a component-wise product. Let $\boldsymbol{u} = [u_1, u_2, \cdots, u_n]^\top$, $\boldsymbol{v} = [v_1, v_2, \cdots, v_n]^\top \in \mathbb{R}^n$,

$$\boldsymbol{u} \odot \boldsymbol{v} := [u_1 v_1, u_2 v_2, \cdots, u_n v_n]^\top.$$

  We can define a diagonal matrix $\boldsymbol{U}$ such that $\boldsymbol{u} \odot \boldsymbol{v} = \boldsymbol{U}\boldsymbol{v}$, where

$$\boldsymbol{U} := \begin{bmatrix} u_1 & 0 & \cdots & 0 \\ 0 & u_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_n \end{bmatrix}.$$

Recalling the CSR problem

CSR problem: *Given* $\boldsymbol{x} \in \mathbb{R}^m$ *and* $\boldsymbol{D} = [\boldsymbol{d}_1, \cdots, \boldsymbol{d}_n] \in \mathbb{R}^{\ell \times n}$*, we seek* $\boldsymbol{Z} = [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n] \in \mathbb{R}^{k \times n}$ *with* $m = \ell + k - 1$ *solving*

$$\min_{\boldsymbol{Z}} \left(\frac{1}{2} \|\boldsymbol{x} - \sum_{j=1}^{n} \boldsymbol{d}_j * \boldsymbol{z}_j\|_2^2 + \lambda \sum_{j=1}^{n} \|\boldsymbol{z}_j\|_1 \right).$$

To solve the above minimization problem, we first use the ADMM algorithm to split it into three subproblems:

$$\begin{aligned}
\boldsymbol{Z}^{(i+1)} &= \arg\min_{\boldsymbol{Z}} \left(\frac{1}{2} \|\boldsymbol{x} - \sum_{j=1}^{n} \boldsymbol{d}_j * \boldsymbol{z}_j\|_2^2 + \frac{\rho}{2} \sum_{j=1}^{n} \|\boldsymbol{z}_j - \boldsymbol{y}_j^{(i)} + \boldsymbol{u}_j^{(i)}\|_2^2 \right), \\
\boldsymbol{Y}^{(i+1)} &= \arg\min_{\boldsymbol{Y}} \left(\lambda \sum_{j=1}^{n} \|\boldsymbol{y}_j\|_1 + \frac{\rho}{2} \sum_{j=1}^{n} \|\boldsymbol{z}_j^{(i+1)} - \boldsymbol{y}_j + \boldsymbol{u}_j^{(i)}\|_2^2, \right) \\
\boldsymbol{U}^{(i+1)} &= \boldsymbol{U}^{(i)} + \boldsymbol{Z}^{(i+1)} - \boldsymbol{Y}^{(i+1)}.
\end{aligned}$$

Using discrete Fourier transform for Z

We will use the discrete Fourier transform and Hadamard product to solve the subproblem of $\boldsymbol{Z}$. We can rewrite these subproblems as

$$\begin{aligned}
\widehat{\boldsymbol{Z}}^{(i+1)} &= \underset{\widehat{\boldsymbol{Z}}}{\arg\min}\left(\frac{1}{2}\|\widehat{\boldsymbol{x}} - \sum_{j=1}^{n} \widehat{\boldsymbol{d}_j'} \odot \widehat{\boldsymbol{z}_j'}\|_2^2 + \frac{\rho}{2}\sum_{j=1}^{n}\|\widehat{\boldsymbol{z}_j'} - \widehat{\boldsymbol{y}_j'}^{(i)} + \widehat{\boldsymbol{u}_j'}^{(i)}\|_2^2\right), \\
\boldsymbol{Y}^{(i+1)} &= \underset{\boldsymbol{Y}}{\arg\min}\left(\lambda\sum_{j=1}^{n}\|\boldsymbol{y}_j\|_1 + \frac{\rho}{2}\sum_{j=1}^{n}\|\mathcal{F}^{-1}(\widehat{\boldsymbol{z}_j'}^{(i+1)}) - \boldsymbol{y}_j + \boldsymbol{u}_j^{(i)}\|_2^2\right), \\
\boldsymbol{U}^{(i+1)} &= \boldsymbol{U}^{(i)} + \mathcal{F}^{-1}(\widehat{\boldsymbol{Z}}^{(i+1)}) - \boldsymbol{Y}^{(i+1)},
\end{aligned}$$

where

$$\mathcal{F}^{-1}(\widehat{\boldsymbol{Z}}) = [\mathcal{F}^{-1}(\widehat{z_1'}), \mathcal{F}^{-1}(\widehat{z_2'}), \cdots, \mathcal{F}^{-1}(\widehat{z_n'})].$$

Why can we use the discrete Fourier transform?

Note that the discrete Fourier transform $\mathcal{F}$ is linear. Thus, we have

$$\begin{aligned}
\frac{1}{2}\|\boldsymbol{x} - \sum_{j=1}^{n} \boldsymbol{d}_j * \boldsymbol{z}_j\|_2^2 &= \frac{1}{2m}\|\mathcal{F}(\boldsymbol{x} - \sum_{j=1}^{n} \boldsymbol{d}_j * \boldsymbol{z}_j)\|_2^2 \quad \text{(Plancherel theorem)} \\
&= \frac{1}{2m}\|\mathcal{F}(\boldsymbol{x}) - \mathcal{F}(\sum_{j=1}^{n} \boldsymbol{d}_j * \boldsymbol{z}_j)\|_2^2 \\
&= \frac{1}{2m}\|\widehat{\boldsymbol{x}} - \sum_{j=1}^{n} \mathcal{F}(\boldsymbol{d}_j * \boldsymbol{z}_j)\|_2^2 = \frac{1}{2m}\|\widehat{\boldsymbol{x}} - \sum_{j=1}^{n} \widehat{\boldsymbol{d}_j'} \odot \widehat{\boldsymbol{z}_j'}\|_2^2.
\end{aligned}$$

Similarly, the second term of subproblem $\boldsymbol{Z}$ can be rewritten as

$$\begin{aligned}
\frac{\rho}{2}\sum_{j=1}^{n}\|\boldsymbol{z}_j - \boldsymbol{y}_j^{(i)} + \boldsymbol{u}_j^{(i)}\|_2^2 &= \frac{\rho}{2}\sum_{j=1}^{n}\|\boldsymbol{z}_j' - \boldsymbol{y}_j'^{(i)} + \boldsymbol{u}_j'^{(i)}\|_2^2 \\
&= \frac{\rho}{2m}\sum_{j=1}^{n}\|\widehat{\boldsymbol{z}_j'} - \widehat{\boldsymbol{y}_j'}^{(i)} + \widehat{\boldsymbol{u}_j'}^{(i)}\|_2^2.
\end{aligned}$$

Note: $\boldsymbol{x} \in \mathbb{R}^m$, $\boldsymbol{d}_j \in \mathbb{R}^\ell$, $\boldsymbol{z}_j \in \mathbb{R}^k$, $\boldsymbol{d}_j * \boldsymbol{z}_j \in \mathbb{R}^{\ell+k-1} = \mathbb{R}^m$, $\boldsymbol{d}_j', \boldsymbol{z}_j' \in \mathbb{R}^m$.

The subproblem of $\widehat{Z}$

We first define

$$\widehat{D_j} = \operatorname{diag}(\widehat{d'_j}). \quad (m \times m \textit{ disgonal matrix})$$

Then the subproblem in the Fourier domain can be posed as:

$$\hat{z}^{(i+1)} = \arg\min_{\hat{z}} \left(\frac{1}{2} \|\hat{x} - \widehat{D}\hat{z}\|_2^2 + \frac{\rho}{2} \|\hat{z} - \hat{y}^{(i)} + \hat{u}^{(i)}\|_2^2 \right),$$

where

$$\widehat{D} = [\widehat{D}_1, \widehat{D}_2, \cdots, \widehat{D}_n]_{m \times mn}, \quad \hat{z} = [\widehat{z'_1}^\top, \widehat{z'_2}^\top, \cdots, \widehat{z'_n}^\top]_{mn \times 1}^\top,$$

and

$$\hat{y} = [\widehat{y'_1}^\top, \widehat{y'_2}^\top, \cdots, \widehat{y'_n}^\top]_{mn \times 1}^\top, \quad \hat{u} = [\widehat{u'_1}^\top, \widehat{u'_2}^\top, \cdots, \widehat{u'_n}^\top]_{mn \times 1}^\top.$$

Note that we drop the scalar factor $1/m$ in the subproblem.

Rewriting the subproblems of ADMM

Using the above definitions, we can rewrite the subproblems of ADMM as:

$$\widehat{\boldsymbol{z}}^{(i+1)} = \min_{\widehat{\boldsymbol{z}}} \left(\frac{1}{2} \| \widehat{\boldsymbol{x}} - \widehat{\boldsymbol{D}} \widehat{\boldsymbol{z}} \|_2^2 + \frac{\rho}{2} \| \widehat{\boldsymbol{z}} - \widehat{\boldsymbol{y}}^{(i)} + \widehat{\boldsymbol{u}}^{(i)} \|_2^2 \right), \quad (13_1)$$

$$\boldsymbol{y}^{(i+1)} = \min_{\boldsymbol{y}} \left(\lambda \| \boldsymbol{y} \|_1 + \frac{\rho}{2} \sum_{j=1}^{n} \| \mathcal{F}^{-1}(\widehat{\boldsymbol{z}}^{(i+1)}) - \boldsymbol{y} + \boldsymbol{u}^{(i)} \|_2^2 \right), \quad (13_2)$$

$$\boldsymbol{u}^{(i+1)} = \boldsymbol{u}^{(i)} + \mathcal{F}^{-1}(\widehat{\boldsymbol{z}}^{(i+1)}) - \boldsymbol{y}^{(i+1)}, \quad (13_3)$$

where

$$\boldsymbol{y} = [\boldsymbol{y}_1'^{\top}, \boldsymbol{y}_2'^{\top}, \cdots, \boldsymbol{y}_n'^{\top}]_{mn \times 1}^{\top}, \quad \boldsymbol{u} = [\boldsymbol{u}_1'^{\top}, \boldsymbol{u}_2'^{\top}, \cdots, \boldsymbol{u}_n'^{\top}]_{mn \times 1}^{\top}.$$

Solving minimization problem (13_1)

First we define

$$F(\widehat{\boldsymbol{z}}) = \frac{1}{2}\|\widehat{\boldsymbol{x}} - \widehat{\boldsymbol{D}}\widehat{\boldsymbol{z}}\|_2^2 + \frac{\rho}{2}\|\widehat{\boldsymbol{z}} - \widehat{\boldsymbol{y}}^{(i)} + \widehat{\boldsymbol{u}}^{(i)}\|_2^2.$$

To solve $\min_{\widehat{\boldsymbol{z}}} F(\widehat{\boldsymbol{z}})$, we compute

$$\begin{aligned}
\nabla F(\widehat{\boldsymbol{z}}) &= -\widehat{\boldsymbol{D}}^\top(\widehat{\boldsymbol{x}} - \widehat{\boldsymbol{D}}\widehat{\boldsymbol{z}}) + \rho \boldsymbol{I}(\widehat{\boldsymbol{z}} - \widehat{\boldsymbol{y}}^{(i)} + \widehat{\boldsymbol{u}}^{(i)}) \\
&= (\widehat{\boldsymbol{D}}^\top\widehat{\boldsymbol{D}} + \rho\boldsymbol{I})\widehat{\boldsymbol{z}} - \left(\widehat{\boldsymbol{D}}^\top\widehat{\boldsymbol{x}} + \rho(\widehat{\boldsymbol{y}}^{(i)} - \widehat{\boldsymbol{u}}^{(i)})\right).
\end{aligned}$$

Letting $\nabla F(\widehat{\boldsymbol{z}}) = \boldsymbol{0}$, we have

$$(\widehat{\boldsymbol{D}}^\top\widehat{\boldsymbol{D}} + \rho\boldsymbol{I})_{mn\times mn}\widehat{\boldsymbol{z}} = \widehat{\boldsymbol{D}}^\top\widehat{\boldsymbol{x}} + \rho(\widehat{\boldsymbol{y}}^{(i)} - \widehat{\boldsymbol{u}}^{(i)}).$$

Therefore, we obtain the solution

$$\widehat{\boldsymbol{z}}^{(i+1)} = (\widehat{\boldsymbol{D}}^\top\widehat{\boldsymbol{D}} + \rho\boldsymbol{I})^{-1}\left(\widehat{\boldsymbol{D}}^\top\widehat{\boldsymbol{x}} + \rho(\widehat{\boldsymbol{y}}^{(i)} - \widehat{\boldsymbol{u}}^{(i)})\right).$$

Solving minimization problem (13_2)

The way to solve minimization problem (13_2) is similar to that for solving problem (6_2).

Finally, we obtain the ADMM iterative scheme as follows:

$$\widehat{\boldsymbol{z}}^{(i+1)} = (\widehat{\boldsymbol{D}}^\top \widehat{\boldsymbol{D}} + \rho \boldsymbol{I})^{-1} (\widehat{\boldsymbol{D}}^\top \widehat{\boldsymbol{x}} + \rho(\widehat{\boldsymbol{y}}^{(i)} - \widehat{\boldsymbol{u}}^{(i)})), \quad (14_1)$$

$$\boldsymbol{y}^{(i+1)} = \mathcal{S}_{\lambda/\rho}(\mathcal{F}^{-1}(\widehat{\boldsymbol{z}}^{(i+1)}) + \boldsymbol{u}^{(i)}), \quad (14_2)$$

$$\boldsymbol{u}^{(i+1)} = \boldsymbol{u}^{(i)} + \mathcal{F}^{-1}(\widehat{\boldsymbol{z}}^{(i+1)}) - \boldsymbol{y}^{(i+1)}. \quad (14_3)$$

Next, we will introduce the *Sherman-Morrison formula* which can be applied to solve $\widehat{\boldsymbol{z}}^{(i+1)}$ in a more efficient way.

The Sherman-Morrison formula: a special case

Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be two $n \times n$ matrices. In general, $(\boldsymbol{A} + \boldsymbol{B})$ is not invertible, even though $\boldsymbol{A}$ is invertible. However, if $\boldsymbol{A}$ is invertible and $\boldsymbol{B}$ has some certain structure, then $(\boldsymbol{A} + \boldsymbol{B})^{-1}$ exists.

A special case of the Sherman-Morrison formula: *Let $\boldsymbol{I}$ be the $n \times n$ identity matrix and $\boldsymbol{u}$, $\boldsymbol{v}$ be two given vectors in $\mathbb{C}^n$. If $1 + \boldsymbol{v}^\top \boldsymbol{u} \neq 0$, then $\boldsymbol{I} + \boldsymbol{u}\boldsymbol{v}^\top$ is invertible and*

$$(\boldsymbol{I} + \boldsymbol{u}\boldsymbol{v}^\top)^{-1} = \boldsymbol{I} - \frac{\boldsymbol{u}\boldsymbol{v}^\top}{1 + \boldsymbol{v}^\top \boldsymbol{u}}.$$

Proof: We check that

$$\begin{aligned}
(\boldsymbol{I} + \boldsymbol{u}\boldsymbol{v}^\top)\left(\boldsymbol{I} - \frac{\boldsymbol{u}\boldsymbol{v}^\top}{1 + \boldsymbol{v}^\top \boldsymbol{u}}\right) &= \boldsymbol{I} - \frac{\boldsymbol{u}\boldsymbol{v}^\top}{1 + \boldsymbol{v}^\top \boldsymbol{u}} + \boldsymbol{u}\boldsymbol{v}^\top - \frac{\boldsymbol{u}\boldsymbol{v}^\top \boldsymbol{u}\boldsymbol{v}^\top}{1 + \boldsymbol{v}^\top \boldsymbol{u}} \\
&= \boldsymbol{I} + \frac{-\boldsymbol{u}\boldsymbol{v}^\top + \boldsymbol{u}\boldsymbol{v}^\top + \boldsymbol{v}^\top \boldsymbol{u}\boldsymbol{u}\boldsymbol{v}^\top}{1 + \boldsymbol{v}^\top \boldsymbol{u}} - \frac{\boldsymbol{u}\boldsymbol{v}^\top \boldsymbol{u}\boldsymbol{v}^\top}{1 + \boldsymbol{v}^\top \boldsymbol{u}} \\
&= \boldsymbol{I} + \frac{\boldsymbol{v}^\top \boldsymbol{u}\boldsymbol{u}\boldsymbol{v}^\top}{1 + \boldsymbol{v}^\top \boldsymbol{u}} - \frac{\boldsymbol{u}\boldsymbol{v}^\top \boldsymbol{u}\boldsymbol{v}^\top}{1 + \boldsymbol{v}^\top \boldsymbol{u}} = \boldsymbol{I}. \quad (\boldsymbol{v}^\top \boldsymbol{u} : \textit{scalar}) \quad \square
\end{aligned}$$

How to derive the inverse?

Given $\boldsymbol{b} \in \mathbb{C}^n$, we consider the linear system $(\boldsymbol{I} + \boldsymbol{u}\boldsymbol{v}^\top)\boldsymbol{x} = \boldsymbol{b}$. Assume that $\boldsymbol{I} + \boldsymbol{u}\boldsymbol{v}^\top$ is invertible. Then the unique solution $\boldsymbol{x}$ exists.
Let $k = \boldsymbol{v}^\top \boldsymbol{x} \in \mathbb{C}$. Then $\boldsymbol{x} + k\boldsymbol{u} = \boldsymbol{b} \;\Rightarrow\; \boldsymbol{v}^\top\boldsymbol{x} + k\boldsymbol{v}^\top\boldsymbol{u} = \boldsymbol{v}^\top\boldsymbol{b}$
$\Rightarrow\; k + k(\boldsymbol{v}^\top\boldsymbol{u}) = \boldsymbol{v}^\top\boldsymbol{b}$, which implies

$$k = \frac{\boldsymbol{v}^\top\boldsymbol{b}}{1 + \boldsymbol{v}^\top\boldsymbol{u}}, \quad \text{if } 1 + \boldsymbol{v}^\top\boldsymbol{u} \neq 0.$$

Therefore, we know that

$$\boldsymbol{x} = \boldsymbol{b} - k\boldsymbol{u} = \boldsymbol{b} - \frac{\boldsymbol{v}^\top\boldsymbol{b}}{1 + \boldsymbol{v}^\top\boldsymbol{u}}\boldsymbol{u} = \boldsymbol{b} - \frac{\boldsymbol{u}\boldsymbol{v}^\top}{1 + \boldsymbol{v}^\top\boldsymbol{u}}\boldsymbol{b} = \overbrace{\left(\boldsymbol{I} - \frac{\boldsymbol{u}\boldsymbol{v}^\top}{1 + \boldsymbol{v}^\top\boldsymbol{u}}\right)}^{\text{inverse of } \boldsymbol{I}+\boldsymbol{u}\boldsymbol{v}^\top}\boldsymbol{b}.$$

The Sherman-Morrison formula: *Suppose that $\boldsymbol{A} \in \mathbb{C}^{n\times n}$ is an invertible matrix and $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{C}^n$. Then $\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^\top$ is invertible if and only if $1 + \boldsymbol{v}^\top\boldsymbol{A}^{-1}\boldsymbol{u} \neq 0$. In this case, we have*

$$(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{v}^\top)^{-1} = \boldsymbol{A}^{-1} - \frac{\boldsymbol{A}^{-1}\boldsymbol{u}\boldsymbol{v}^\top\boldsymbol{A}^{-1}}{1 + \boldsymbol{v}^\top\boldsymbol{A}^{-1}\boldsymbol{u}}.$$

How to compute (14_1)?

Recall that

$$(\widehat{\boldsymbol{D}}^{\top}\widehat{\boldsymbol{D}} + \rho \boldsymbol{I})\widehat{\boldsymbol{z}}^{(i+1)} = (\widehat{\boldsymbol{D}}^{\top}\widehat{\boldsymbol{x}} + \rho(\widehat{\boldsymbol{y}}^{(i)} - \widehat{\boldsymbol{u}}^{(i)})), \quad (14_1)$$

where matrix $\widehat{\boldsymbol{D}}$ has the following structure:

$$\begin{aligned}
\widehat{\boldsymbol{D}} &= [\widehat{\boldsymbol{D}}_1, \widehat{\boldsymbol{D}}_2, \cdots, \widehat{\boldsymbol{D}}_n]_{m\times mn} \\
&= \begin{bmatrix}
\widehat{d'_{1,1}} & 0 & \cdots & 0 & \widehat{d'_{2,1}} & 0 & \cdots & 0 & \cdots \\
0 & \widehat{d'_{1,2}} & \ddots & \vdots & 0 & \widehat{d'_{2,2}} & \ddots & \vdots & \cdots \\
\vdots & \ddots & \ddots & 0 & \vdots & \ddots & \ddots & 0 & \cdots \\
0 & \cdots & 0 & \widehat{d'_{1,m}} & 0 & \cdots & 0 & \widehat{d'_{2,m}} & \cdots
\end{bmatrix},
\end{aligned}$$

where

$$\widehat{\boldsymbol{D}_j} = \operatorname{diag}(\widehat{\boldsymbol{d}'_j}) \quad (m \times m \textit{ diagonal matrix}).$$

The structure of matrix $\widehat{\boldsymbol{D}}^\top \widehat{\boldsymbol{D}} + \rho \boldsymbol{I}$

Note that

$$
\begin{aligned}
\widehat{\boldsymbol{D}}^\top \widehat{\boldsymbol{D}} + \rho \boldsymbol{I} &= \begin{bmatrix} \widehat{\boldsymbol{D}}_1^\top \\ \widehat{\boldsymbol{D}}_2^\top \\ \vdots \\ \widehat{\boldsymbol{D}}_n^\top \end{bmatrix} [\widehat{\boldsymbol{D}}_1, \widehat{\boldsymbol{D}}_2, \cdots, \widehat{\boldsymbol{D}}_n] + \rho \boldsymbol{I} \\
&= \begin{bmatrix} \widehat{\boldsymbol{D}}_1^\top \widehat{\boldsymbol{D}}_1 + \rho \boldsymbol{I}_m & \widehat{\boldsymbol{D}}_1^\top \widehat{\boldsymbol{D}}_2 & \cdots & \widehat{\boldsymbol{D}}_1^\top \widehat{\boldsymbol{D}}_n \\ \widehat{\boldsymbol{D}}_2^\top \widehat{\boldsymbol{D}}_1 & \widehat{\boldsymbol{D}}_2^\top \widehat{\boldsymbol{D}}_2 + \rho \boldsymbol{I}_m & \cdots & \widehat{\boldsymbol{D}}_2^\top \widehat{\boldsymbol{D}}_n \\ \vdots & \vdots & \ddots & \vdots \\ \widehat{\boldsymbol{D}}_n^\top \widehat{\boldsymbol{D}}_1 & \widehat{\boldsymbol{D}}_n^\top \widehat{\boldsymbol{D}}_2 & \cdots & \widehat{\boldsymbol{D}}_n^\top \widehat{\boldsymbol{D}}_n + \rho \boldsymbol{I}_m \end{bmatrix}_{mn \times mn}.
\end{aligned}
$$

By re-ordering the equations, the $mn \times mn$ system $\left(\widehat{\boldsymbol{D}}^\top \widehat{\boldsymbol{D}} + \rho \boldsymbol{I}\right) \widehat{\boldsymbol{z}}^{(i+1)} = \boldsymbol{R}$ can be replaced by m independent linear systems of size $n \times n$, each of which consists of a rank one component plus a diagonal component, then solved by the Sherman-Morrison formula, see [Wohlberg 2016, Appendix A].

Convolutional sparse dictionary learning problem

We now consider the convolutional sparse dictionary learning problem, where the dictionary $\boldsymbol{D}$ is unknown and needed to be sought together with the convolutional sparse solution.

Convolutional SDL problem: *Let* $\{\boldsymbol{x}_i\}_{i=1}^N \subset \mathbb{R}^m$ *be a given dataset of signals. We seek a dictionary matrix* $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_M] \in \mathbb{R}^{\ell \times M}$ *and the coefficient matrices* $\{\boldsymbol{Z}_i\}_{i=1}^N \subset \mathbb{R}^{k \times M}$ *with* $\boldsymbol{Z}_i = [\boldsymbol{z}_{1,i}, \boldsymbol{z}_{2,i}, \cdots, \boldsymbol{z}_{M,i}]$ *and* $m = \ell + k - 1$ *such that* $\boldsymbol{D}$ *and* $\{\boldsymbol{Z}_i\}_{i=1}^N$ *solve the following minimization problem:*

$$\min_{\{\boldsymbol{d}_j\}_{j=1}^M, \{\boldsymbol{z}_{j,i}\}_{j=1,i=1}^{M,N}} \left(\frac{1}{2} \sum_{i=1}^N \|\boldsymbol{x}_i - \sum_{j=1}^M \boldsymbol{d}_j * \boldsymbol{z}_{j,i}\|_2^2 + \lambda \sum_{i=1}^N \sum_{j=1}^M \|\boldsymbol{z}_{j,i}\|_1 \right)$$

$$\text{subject to } \|\boldsymbol{d}_j\|_2 \leq 1 \quad \forall\, j = 1, 2, \cdots, M,$$

where $\lambda > 0$ *is a given penalty parameter.*

Toeplitz matrix

- Define $\widetilde{\boldsymbol{D}} = \begin{bmatrix} \boldsymbol{D}_1 & \boldsymbol{D}_2 & \cdots & \boldsymbol{D}_M \end{bmatrix}$ with each $\boldsymbol{D}_j$ is the *Toeplitz matrix* defined with respect to $\boldsymbol{d}_j$ as before.

  Define $\boldsymbol{z}_i = \begin{bmatrix} \boldsymbol{z}_{1,i}^\top & \boldsymbol{z}_{2,i}^\top & \cdots & \boldsymbol{z}_{M,i}^\top \end{bmatrix}^\top$ for $i = 1, 2, \cdots, N$, where each $\boldsymbol{z}_i$ is the coefficient vector with respect to the data $\boldsymbol{x}_i$.

  Define $\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{z}_1 & \boldsymbol{z}_2 & \cdots & \boldsymbol{z}_N \end{bmatrix}$ and $\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_N \end{bmatrix}$.

- The convolutional SDL problem can be simplified as

$$\min_{\widetilde{\boldsymbol{D}},\, \boldsymbol{Z}} \left(\frac{1}{2} \|\boldsymbol{X} - \widetilde{\boldsymbol{D}}\boldsymbol{Z}\|_F^2 + \lambda \|\boldsymbol{Z}\|_{1,1} \right)$$

$$\text{subject to } \|\boldsymbol{d}_j\|_2 \le 1 \quad \forall\, j = 1, 2, \cdots, M,$$

  where $\|\boldsymbol{Z}\|_{1,1}$ is defined as before.

How to solve the convolutional SDL problem?

Though we can still use the ADMM iterative scheme to solve

$$\min_{\widetilde{D},\, Z} \left(\frac{1}{2} \|X - \widetilde{D} Z\|_F^2 + \lambda \|Z\|_{1,1} \right)$$
$$\text{subject to } \|d_j\|_2 \le 1 \quad \forall\, j = 1, 2, \cdots, M,$$

the sizes of the involved matrices are too large. Thus, we will use the DFT and the Sherman-Morrison formula to deal with this problem. The steps are similar to the CSR problem, but more complicated.

Recall the convolutional SDL problem:

$$\min_{\{d_j\}_{j=1}^M, \{z_{j,i}\}_{j=1,i=1}^{M,N}} \left(\frac{1}{2} \sum_{i=1}^N \|x_i - \sum_{j=1}^M d_j * z_{j,i}\|_2^2 + \lambda \sum_{i=1}^N \sum_{j=1}^M \|z_{j,i}\|_1 \right)$$
$$\text{subject to } \|d_j\|_2 \le 1 \quad \forall\, j = 1, 2, \cdots, M.$$

For solving this problem, we split it into two parts.

Step 1: Solving the coefficient Z

For solving the convolutional SDL problem, we first give an initial dictionary $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_M]$ to solve coefficient $\boldsymbol{Z}$ and then further use ADMM algorithm to split this problem into three subproblems:

$$\widehat{\boldsymbol{Z}}^{(i+1)} = \arg\min_{\widehat{\boldsymbol{Z}}} \Big(\frac{1}{2} \sum_{i=1}^{N} \|\widehat{\boldsymbol{x}}_i - \sum_{j=1}^{M} \widehat{\boldsymbol{d}'_j} \odot \widehat{\boldsymbol{z}'_{j,i}}\|_2^2 + \frac{\rho}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} \|\widehat{\boldsymbol{z}'_{j,i}} - \widehat{\boldsymbol{y}'_{j,i}}^{(i)} + \widehat{\boldsymbol{u}'_{j,i}}^{(i)}\|_2^2 \Big)$$

$$\boldsymbol{Y}^{(i+1)} = \arg\min_{\boldsymbol{Y}} \Big(\lambda \sum_{i=1}^{N} \sum_{j=1}^{M} \|\boldsymbol{y}_{j,i}\|_1 + \frac{\rho}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} \|\mathcal{F}^{-1}(\widehat{\boldsymbol{z}'_{j,i}}) - \boldsymbol{y}_{j,i} + \boldsymbol{u}'_{j,i}\|_2^2 \Big),$$

$$\boldsymbol{U}^{(i+1)} = \boldsymbol{U}^{(i)} + \mathcal{F}^{-1}(\widehat{\boldsymbol{Z}}^{(i+1)}) - \boldsymbol{Y}^{(i+1)},$$

with

$$\mathcal{F}^{-1}(\widehat{\boldsymbol{Z}}) = \begin{bmatrix} \mathcal{F}^{-1}(\widehat{\boldsymbol{z}'_{1,1}}) & \cdots & \mathcal{F}^{-1}(\widehat{\boldsymbol{z}'_{1,M}}) \\ \vdots & \ddots & \vdots \\ \mathcal{F}^{-1}(\widehat{\boldsymbol{z}'_{N,1}}) & \cdots & \mathcal{F}^{-1}(\widehat{\boldsymbol{z}'_{N,M}}) \end{bmatrix}.$$

Rewritten in a compact form

For convenience, we can rewrite these subproblems as follows:

$$
\begin{aligned}
\widehat{\boldsymbol{Z}}^{(i+1)} &= \arg\min_{\widehat{\boldsymbol{Z}}} \left(\frac{1}{2}\|\widehat{\boldsymbol{X}} - \widehat{\boldsymbol{D}}\widehat{\boldsymbol{Z}}\|_F^2 + \frac{\rho}{2}\|\widehat{\boldsymbol{Z}} - \widehat{\boldsymbol{Y}}^{(i)} + \widehat{\boldsymbol{U}}^{(i)}\|_F^2 \right), \\
\boldsymbol{Y}^{(i+1)} &= \arg\min_{\boldsymbol{Y}} \left(\lambda\|\boldsymbol{Y}\|_{1,1} + \frac{\rho}{2}\|\mathcal{F}^{-1}(\widehat{\boldsymbol{Z}}^{(i+1)}) - \widehat{\boldsymbol{Y}} + \widehat{\boldsymbol{U}}^{(i)}\|_F^2 \right), \\
\boldsymbol{U}^{(i+1)} &= \boldsymbol{U}^{(i)} + \mathcal{F}^{-1}(\widehat{\boldsymbol{Z}}^{(i+1)}) - \boldsymbol{Y}^{(i+1)},
\end{aligned}
$$

with

$$
\widehat{\boldsymbol{X}} = [\widehat{\boldsymbol{x}_1}, \widehat{\boldsymbol{x}_2}, \cdots, \widehat{\boldsymbol{x}_N}], \quad \widehat{\boldsymbol{D}} = [\widehat{\boldsymbol{D}_1}, \widehat{\boldsymbol{D}_2}, \cdots, \widehat{\boldsymbol{D}_M}],
$$

and

$$
\boldsymbol{Y} = \begin{bmatrix} \boldsymbol{y}'_{1,1} & \cdots & \boldsymbol{y}'_{1,M} \\ \vdots & \ddots & \vdots \\ \boldsymbol{y}'_{N,1} & \cdots & \boldsymbol{y}'_{N,M} \end{bmatrix}, \quad \boldsymbol{U} = \begin{bmatrix} \boldsymbol{u}'_{1,1} & \cdots & \boldsymbol{u}'_{1,M} \\ \vdots & \ddots & \vdots \\ \boldsymbol{u}'_{N,1} & \cdots & \boldsymbol{u}'_{N,M} \end{bmatrix}.
$$

Using the similar ways as that for solving CSR problem, we can solve the above subproblems.

Step 2: Solving the dictionary D

Recall the convolutional SDL problem:

$$\min_{\{\boldsymbol{d}_j\}_{j=1}^M,\ \{\boldsymbol{z}_{j,i}\}_{j=1,i=1}^{M,N}} \left(\frac{1}{2} \sum_{i=1}^N \|\boldsymbol{x}_i - \sum_{j=1}^M \boldsymbol{d}_j * \boldsymbol{z}_{j,i}\|_2^2 + \lambda \sum_{i=1}^N \sum_{j=1}^M \|\boldsymbol{z}_{j,i}\|_1 \right)$$
$$\text{subject to } \|\boldsymbol{d}_j\|_2 \le 1 \quad \forall\, j = 1, 2, \cdots, M.$$

When the coefficient $\boldsymbol{Z}$ is obtained, the blue term is a given number. Solving the dictionary $\boldsymbol{D}$ is equivalent to solve

$$\min_{\{\boldsymbol{d}_j\}_{j=1}^M} \frac{1}{2} \sum_{i=1}^N \|\boldsymbol{x}_i - \sum_{j=1}^M \boldsymbol{d}_j * \boldsymbol{z}_{j,i}\|_2^2 \text{ subject to } \|\boldsymbol{d}_j\|_2 \le 1, \quad \forall\, j = 1, 2, \cdots, M.$$

Using ADMM algorithm to solve Step 2

We use the ADMM algorithm to solve the above problem:

$$
\begin{aligned}
\boldsymbol{D}^{(i+1)} &= \arg\min_{\boldsymbol{D}} \Big(\frac{1}{2} \sum_{i=1}^{N} \| \boldsymbol{x}_i - \sum_{j=1}^{M} \boldsymbol{d}_j * \boldsymbol{z}_j \|_2^2 + \frac{\rho}{2} \sum_{j=1}^{M} \| \boldsymbol{d}_j - \boldsymbol{g}_j^{(i)} + \boldsymbol{h}_j^{(i)} \|_2^2 \Big), \\
\boldsymbol{G}^{(i+1)} &= \mathrm{proj}_{g_{(\boldsymbol{G})}} \{ \boldsymbol{D}^{(i+1)} \}, \\
\boldsymbol{H}^{(i+1)} &= \boldsymbol{H}^{(i)} + \boldsymbol{D}^{(i+1)} - \boldsymbol{G}^{(i+1)},
\end{aligned}
$$

and then use the Fourier transform and similar ways as before,

$$
\begin{aligned}
\widehat{\boldsymbol{D}}^{(i+1)} &= \arg\min_{\widehat{\boldsymbol{D}}} \Big(\frac{1}{2} \sum_{i=1}^{N} \| \widehat{\boldsymbol{x}}_i - \sum_{j=1}^{M} \widehat{\boldsymbol{d}}_j' \odot \widehat{\boldsymbol{z}}_j' \|_2^2 + \frac{\rho}{2} \sum_{j=1}^{M} \| \widehat{\boldsymbol{d}}_j' - \widehat{\boldsymbol{g}}_j'^{(i)} + \widehat{\boldsymbol{h}}_j'^{(i)} \|_2^2 \Big), \\
\boldsymbol{G}^{(i+1)} &= \mathrm{proj}_{g_{(\boldsymbol{G})}} \{ \mathcal{F}^{-1}(\widehat{\boldsymbol{D}}^{(i+1)}) \}, \\
\boldsymbol{H}^{(i+1)} &= \boldsymbol{H}^{(i)} + \mathcal{F}^{-1}(\widehat{\boldsymbol{D}}^{(i+1)}) - \boldsymbol{G}^{(i+1)},
\end{aligned}
$$

where

$$
\mathcal{F}^{-1}(\widehat{\boldsymbol{D}}) = [\mathcal{F}^{-1}(\widehat{\boldsymbol{d}_1'}), \mathcal{F}^{-1}(\widehat{\boldsymbol{d}_2'}), \cdots, \mathcal{F}^{-1}(\widehat{\boldsymbol{d}_M'})].
$$

References

[1] H. Bristow, A. Eriksson, and S. Lucey, Fast convolutional sparse coding, *2013 IEEE CVPR,* pp. 391-398.

[2] F. Heide, W. Heidrich, and G. Wetzstein, Fast and flexible convolutional sparse coding, *2015 IEEE CVPR,* pp. 5135-5143.

[3] H. Sreter and R. Giryes, Learned convolutional sparse coding, *2018 IEEE ICASSP,* pp. 2191-2195.

[4] B. Wohlberg, Efficient convolutional sparse coding, *2014 IEEE ICASSP,* pp. 7173-7177.

[5] B. Wohlberg, Efficient algorithms for convolutional sparse representations, *IEEE Transactions on Image Processing,* 25 (2016), pp. 301-315.

[6] J. Xiong , P. Richtarik, and W. Heidrich, Stochastic convolutional sparse coding, *Vision, Modeling, and Visualization,* 2019.