Note for GSR: Group-Based Sparse Representation for Image Restoration

Yao Zhang

1.1 Introduction

It is a typical ill-posed linear inverse problem and can be generally formulated as:

$$\boldsymbol{y} = H\boldsymbol{x} + \boldsymbol{n} \tag{1.1}$$

where $\boldsymbol{x}, \boldsymbol{y}$ are lexicographically stacked representations of the original image and the degraded image, respectively, H is a matrix representing a non-invertible linear degradation operator and $\boldsymbol{n}$ is usually additive Gaussian white noise.

To cope with the ill-posed nature of image restoration, image prior knowledge is usually employed for regularizing the solution to the following minimization problem

$$\arg\min_{\boldsymbol{x}} \frac{1}{2}\left\|H\boldsymbol{x} - \boldsymbol{y}\right\|_2^2 + \lambda\Psi\left(\boldsymbol{x}\right) \tag{1.2}$$

where $\frac{1}{2}\left\|H\boldsymbol{x} - \boldsymbol{y}\right\|_2^2$ is l_2 data-fidelity term, $\Psi\left(\boldsymbol{x}\right)$ is called the regularization term denoting image prior and λ is the regularization parameter.

1.2 Traditional Patch-based Sparse Representation

The basic unit of sparse representation for natural image is patch. Mathematically, denote by $\boldsymbol{x} \in \mathbb{R}^N$ and $\boldsymbol{x}_k \in \mathbb{R}^{\mathcal{B}_s}$ the vector representations of the original image and an image patch of size $\sqrt{\mathcal{B}_s} \times \sqrt{\mathcal{B}_s}$ at location k. Then

$$\boldsymbol{x}_k = R_k\left(\boldsymbol{x}\right) \tag{1.3}$$

where $R_k\left(\cdot\right)$ is an operator that extracts the patch $\boldsymbol{x}_k$ from the image $\boldsymbol{x}$, and transpose, denoted by $R_k^T\left(\cdot\right)$, is able to put back a patch into the k-th position in the reconstructed image. Considering that patches are usually overlapped, the recovery of $\boldsymbol{x}$ from $\{x_k\}$ becomes

$$\boldsymbol{x} = \sum_{k=1}^{n} R_k^T\left(\boldsymbol{x}_k\right). \Big/ \sum_{k=1}^{n} R_k^T\left(1_{\mathcal{B}_s}\right) \tag{1.4}$$

where the notion ./ stands for the element-wise division of two vectors, and $1_{\mathcal{B}_s}$ is a vector of size $\mathcal{B}_s$ with all its elements being 1. For a given dictionary $D \in \mathbb{R}^{\mathcal{B}_s \times M}$ (M is the number of atoms in D), the sparse coding process of each patch $\boldsymbol{x}_k$ over D is to find a sparse vector $\alpha_k \in \mathbb{R}^M$ (i.e. most of the coefficients in α_k are zero or close to zero) such that $\boldsymbol{x}_k \approx D\alpha_k$. Then the entire image can be sparsely represented by the set of sparse codes $\{\alpha_k\}$. In practice,

$$\alpha_k = \arg\min_{\alpha} \frac{1}{2}\left\|\boldsymbol{x}_k - D\alpha\right\|_2^2 + \lambda\|\alpha\|_p \tag{1.5}$$

where $\alpha_k \in \mathbb{R}^M$.

Similar to Eq 1.4, reconstructing $\boldsymbol{x}$ from its sparse codes $\{\alpha_k\}$ is formulated:

$$\boldsymbol{x} = D \circ \alpha \triangleq \sum_{k=1}^{n} R_k^T \left(D\alpha_k\right) ./ \sum_{k=1}^{n} R_k^T \left(1_{\mathcal{B}_s}\right) \tag{1.6}$$

where $\alpha = \left[\alpha_1^T, \alpha_2^T, ..., \alpha_n^T\right]^T \in \mathbb{R}^{Mn\times 1}$.

Now, considering the degraded version in Eq 1.1, the regularization-based image restoration scheme utilizing traditional patch-based sparse representation model is formulated as

$$\widehat{\alpha} = \arg\min_{\alpha} \frac{1}{2} \|HD \circ \alpha - \boldsymbol{y}\|_2^2 + \lambda \|\alpha\|_p \tag{1.7}$$

With $\widehat{\alpha}$, the reconstructed image can be expressed by $\widehat{x} = D \circ \widehat{\alpha}$.

The heart of the sparse representation modeling lies in the choice of dictionary D. In other words, how seek the best domain to sparsity a given image? Much effort has been devoted to learning a redundant dictionary from a set of training example image patches. To be concrete, given a set of training patches $X = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_J]$, where J is the number of training image patches, the goal of dictionary learning is to jointly optimize the dictionary D and the representation coefficients matrix $\Lambda = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_J]$ such that $\boldsymbol{x}_k \approx D\boldsymbol{\alpha}_k$ and $\|\boldsymbol{\alpha}_k\|_p \leqslant L$.

This can be formulated by the following minimization problem

$$\left(\widehat{D}, \widehat{\Lambda}\right) = \arg\min_{D,\Lambda} \sum_{k=1}^{J} \|\boldsymbol{x}_k - D\boldsymbol{\alpha}_k\|_2^2 \quad s.t. \ \|\boldsymbol{\alpha}_k\|_p \leqslant L, \ \forall k. \tag{1.8}$$

Apparently, 1.8 is large-scale and highly non-convex even when p is 1. To make it tractable and solvable, some approximation approaches including MOD and K-SVD, have been proposed to optimize D and Γ alternatively, leading to many state-of-the-art results in image processing.

However, these approximation approaches for dictionary learning inevitably require high computational complexity.

1.3 Group-Based Sparse Representation

[1] propose a novel spare representation modeling in the unit of group instead of patch, aiming to exploit the local sparsity and nonlocal self-similarity of natural images simultaneously in a unified framework. Each group is represented by the form of matrix, which is composed of nonlocal patches with similar structures.

Group Construction

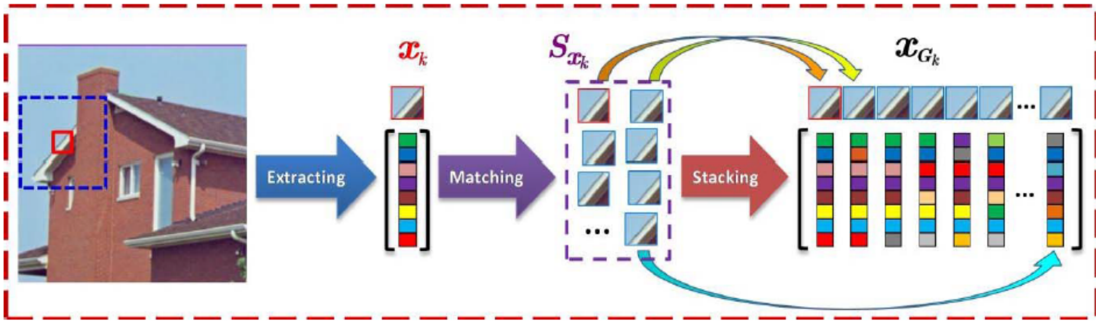

Figure 1: Illustrations for the group construction. Extract each patch $\boldsymbol{x}_k$ from image $\boldsymbol{x}$. For each $\boldsymbol{x}_k$, denote $S_{\boldsymbol{x}_k}$ the set composed of its c best matched patches. Stack all the patches in $S_{\boldsymbol{x}_k}$ in the form of matrix to construct the group, denoted by $\boldsymbol{x}_{G_k}$.

As shown in Fig 1, first, divide the image $\boldsymbol{x}$ with size N into n overlapped patches of size $\sqrt{\mathcal{B}_s} \times \sqrt{\mathcal{B}_s}$ and each patches is denoted by the vector $\boldsymbol{x}_k \in \mathbb{R}^{\mathcal{B}_s}$, i.e. $k = 1, 2, ..., n$.

Then, for each patch $\boldsymbol{x}_k$, denoted by small red square in Fig 1, in the $L \times L$ training window(big blue square), search its c best matched patches, which comprise the set $S_{\boldsymbol{x}_k}$. Here, Euclidean distance is selected as the similarity criterion between different patches.

Next, all the in $S_{\boldsymbol{x}_k}$ are stacked into a matrix of size $\mathcal{B}_s \times c$, denoted by $\boldsymbol{x}_{G_k} \in \mathbb{R}^{\mathcal{B}_s \times c}$, which is includes every patch in $S_{\boldsymbol{x}_k}$ as its columns, i.e. $\boldsymbol{x}_{G_k} = \{\boldsymbol{x}_{G_k \otimes 1}, \boldsymbol{x}_{G_k \otimes 2,...,}\boldsymbol{x}_{G_k \otimes c}\}$. The matrix $\boldsymbol{x}_{G_k}$ containing all the patches with similar structures is named as a group. Similarly to Eq 1.3, we define

$$x_{G_k} = R_{G_k}(x) \tag{1.9}$$

where $R_{G_k}(\cdot)$ is actually an operator that extracts the group $\boldsymbol{x}_{G_k}$ from $\boldsymbol{x}$, and transpose, denote by $R_{G_k}^T(\cdot)$, can put back a group into the k-th position in the reconstructed image padded with zeros elsewhere.

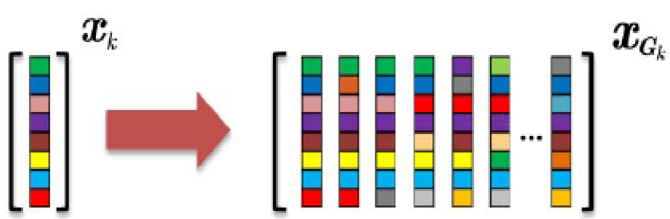

Figure 2: Comparison between patch $\boldsymbol{x}_k$ and $\boldsymbol{x}_{G_k}$. One can also see that the construction of $\boldsymbol{x}_{G_k}$ explicitly exploits the self-similarity of natural images.

By averaging all the groups, the recovery of the whole image $\boldsymbol{x}$ from $\{\boldsymbol{x}_{G_k}\}$ becomes

$$\boldsymbol{x} = \sum_{k=1}^{n} R_{G_k}^T(\boldsymbol{x}_{G_k})./\sum_{k=1}^{n} R_{G_k}^T(1_{\mathcal{B}_s \times c}) \tag{1.10}$$

where ./ stands for the element-wise division of two vectors and $1_{\mathcal{B}_s \times c}$ is a matrix of size $\mathcal{B}_s \times c$ with all the elements being 1.

Group-Based Sparse Representation Modeling

The proposed group-based sparse representation(GSR) model assume that each group $\boldsymbol{x}_{G_k}$ can be accurately represented by a few atoms of a self-adaptive learning dictionary D_{G_k}.

$D_{G_k} = [d_{G_k \otimes 1}, d_{G_k \otimes 2}, \ldots, d_{G_k \otimes m}]$ is supposed to be shown, where $d_{G_k \otimes \ i} \in \mathbb{R}^{\mathcal{B}_s \times c}$ is a matrix of the same size as the group $\boldsymbol{x}_{G_k}$, and m is the number of atoms in D_{G_k}. Different from the dictionary in patch sparse representation $D \in \mathbb{R}^{\mathcal{B}_s \times M}$, here $D_{G_k} \in \mathbb{R}^{(\mathcal{B}_s \times c) \times m}$.

The sparse coding process of each group $\boldsymbol{x}_{G_k}$ over D_{G_k} is to seek a sparse vector $\alpha_{G_k} = [\alpha_{G_k \otimes 1}, \alpha_{G_k \otimes 2}, , \ldots, \alpha_{G_k \otimes m}]$ such that

$$\underbrace{\boldsymbol{x}_{G_k}}_{\in \mathbb{R}^{\mathcal{B}_s \times c}} \approx \sum_{i=1}^{m} \underbrace{\alpha_{G_k \otimes i}}_{\in \mathbb{R}^{1 \times 1}} \times \underbrace{d_{G_k \otimes i}}_{\in \mathbb{R}^{\mathcal{B}_s \times c}} \tag{1.11}$$

and denote that

$$D_{G_k} \alpha_{G_k} \triangleq \sum_{i=1}^{m} \alpha_{G_k \otimes i} d_{G_k \otimes i} \tag{1.12}$$

Then the entire image can be sparsely represented by the set of sparse codes $\{\alpha_{G_k}\}$ in the group domain. Reconstructing $\boldsymbol{x}$ from the sparse codes $\{\alpha_{G_k}\}$ is expressed as

$$\boldsymbol{x} = D_{G_G} \circ \alpha_G \triangleq \sum_{k=1}^{n} R_{G_k}^T \left(D_{G_k} \alpha_{G_k} \right) ./ \sum_{k=1}^{n} R_{G_k}^T \left(1_{\mathcal{B}_s \times c} \right) \tag{1.13}$$

where D_G denotes the concatenation of all α_{G_k}, and denotes the concatenation of all α_{G_k}.

Back to Eq 1.10.

By considering the degraded version in Eq 1.1, the proposed regularization-based image restoration scheme via GSR is formulated as

$$\widehat{\alpha}_G = \arg\min_{\alpha_G} \frac{1}{2} \left\| H D_G \circ \alpha_G - \boldsymbol{y} \right\|_2^2 + \lambda \|\alpha_G\|_0 \tag{1.14}$$

We can see the differences between Eq 1.14 and Eq 1.7 lie in the dictionary and the unit of sparse representation.

Self-Adaptive Group Dictionary Learning

[1] will show how to learn the adaptive dictionary D_{G_k} for each group $\boldsymbol{x}_{G_k}$. On one hand, we hope that each $\boldsymbol{x}_{G_k}$ can be faithfully represented by D_{G_k}. On the other hand, it is expected that the representation coefficient vector of $\boldsymbol{x}_{G_k}$ over D_{G_k} is as sparse as possible.

The adaptive dictionary learning of group can be intuitively formulated as:

$$\arg\min_{D_{\boldsymbol{x}}, \alpha_{G_k}} \sum_{k=1}^{n} \|\boldsymbol{x}_{G_k} - D_{\boldsymbol{x}} \alpha_{G_k}\|_2^2 + \lambda \sum_{k=1}^{n} \|\alpha_{G_k}\|_p \tag{1.15}$$

Eq 1.15 is a joint optimization problem of $D_{\boldsymbol{x}}$ and $\{\alpha_{G_k}\}$, which can be solved by alternatively optimizing $D_{\boldsymbol{x}}$ and $\{\alpha_{G_k}\}$.

Remark 1.1. *It is $D_{\boldsymbol{x}}$ in Eq 1.15, not D_{G_k}.*

[1], utilized D_{G_k} instead of $D_{\boldsymbol{x}}$ based on the following three considerations.

1. Solving the joint optimization in Eq 1.15 requires much computational cost.
2. Eq 1.15 is actually adaptive for given image $\boldsymbol{x}$, not adaptive for a group $\boldsymbol{x}_{G_k}$.
3. Eq 1.15 neglects the characteristics of each group $\boldsymbol{x}_{G_k}$.

So,

$$\arg\min_{D_{\boldsymbol{x}_{G_k}}, \alpha_{G_k}} \sum_{k=1}^{n} \left\|\boldsymbol{x}_{G_k} - D_{\boldsymbol{x}_{G_k}} \alpha_{G_k}\right\|_2^2 + \lambda \sum_{k=1}^{n} \|\alpha_{G_k}\|_p \tag{1.16}$$

We propose to learn the adaptive dictionary D_{G_k} for each group $\boldsymbol{x}_{G_k}$ directly from its estimate r_{G_k} (see Eq 1.29).

After obtaining r_{G_k}, we then apply SVD to it,

$$r_{G_k} = U_{G_k} \sum_{G_k} V_{G_k}^T = \sum_{i=1}^{m} \gamma_{r_{G_k \otimes i}} \left(u_{G_k \otimes i} v_{G_k \otimes i}^T\right) \tag{1.17}$$

Remark 1.2. *Recall the SVD:*

$$
\begin{aligned}
\underbrace{X}_{\in\mathbb{R}^{m\times n}} &= \underbrace{U}_{\in\mathbb{R}^{m\times m}} \underbrace{\sum}_{\in\mathbb{R}^{m\times n}} \underbrace{V^T}_{\in\mathbb{R}^{n\times n}} \\
&= \left[\underbrace{u_1}_{\in\mathbb{R}^{m\times 1}}, u_2, \dots, u_m \right] \begin{bmatrix} \sigma_{11} & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \cdots & 0 \\ 0 & \cdots & \sigma_{rr} & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \underbrace{v_1^T}_{\in\mathbb{R}^{1\times m}} \\ v_2^T \\ \vdots \\ v_n^T \end{bmatrix} \\
&= \sum_{i=1}^{\min\{m,n\}} \sigma_{ii} u_i v_i^T \\
&= \sum_{i=1}^{r} \sigma_{ii} u_i v_i^T
\end{aligned} \tag{1.18}
$$

So, each atom in D_{G_k} for group $\boldsymbol{x}_{G_k}$, is defined as

$$d_{G_k \otimes i} = u_{G_k \otimes i} v_{G_k \otimes i}^T, \quad 1 \leqslant i \leqslant m \tag{1.19}$$

where $d_{G_k \otimes i} \in \mathbb{R}^{\mathcal{B}_s \times c}$, then

$$D_{G_k} = \left[d_{G_k \otimes 1}, d_{G_k \otimes 2}, \dots, d_{G_k \otimes m} \right]. \tag{1.20}$$

1.4 Optimization for GSR-Driven l_0 Minimization

The straightforward method to solve Eq 1.14 is translated into solving l_1 convex form, i.e.

$$\widehat{\alpha}_G = \arg\min_{\alpha_G} \frac{1}{2} \| H D_G \circ \alpha_G - \boldsymbol{y} \|_2^2 + \lambda \| \alpha_G \|_1 \tag{1.21}$$

But in this paper [1], adopts the framework of split Bregman iteration(SBI) [2] to solve Eq 1.14.

But $\| \cdot \|_0$ is not convex !!!

Then first of all, let's make a brief review of SBI. Consider a constrained optimization problem

$$\min_{\boldsymbol{u}\in\mathbb{R}^N, \boldsymbol{v}\in\mathbb{R}^M} f(\boldsymbol{u}) + g(\boldsymbol{v}), \ s.t. \ \underbrace{\boldsymbol{u} = G\boldsymbol{v}}_{\Leftrightarrow \|\boldsymbol{u} - G\boldsymbol{v}\|_2^2 = 0} \tag{1.22}$$

where $G \in \mathbb{R}^{M\times N}$ and $f : \mathbb{R}^N \to \mathbb{R}$, $g : \mathbb{R}^M \to \mathbb{R}$ are convex functions. The SBI solve Eq 1.22 as Algorithm 1.

Algorithm 1 Split Bregman Iteration(SBI)

Set $t = 0$, choose $\mu > 0, \boldsymbol{b}_0 = \mathbf{0}, \boldsymbol{u}_0 = \mathbf{0}, \boldsymbol{v}_0 = \mathbf{0}$.
Repeat

$\boldsymbol{u}^{t+1} = \arg\min_{\boldsymbol{u}} \ f(\boldsymbol{u}) \ + \ \frac{\mu}{2} \left\|\boldsymbol{u} - G\boldsymbol{v}^t - b^t\right\|_2^2$

$\boldsymbol{v}^{t+1} = \arg\min_{\boldsymbol{v}} \ g(\boldsymbol{v}) \ + \ \frac{\mu}{2} \left\|\boldsymbol{u}^{t+1} - G\boldsymbol{v} - b^t\right\|_2^2$

$\boldsymbol{b}^{t+1} = \boldsymbol{b}^t - \left(\boldsymbol{u}^{t+1} - G\boldsymbol{v}^{t+1}\right)$

$t \leftarrow t + 1$

Until stopping criterion satisfied

Remark 1.3. *In Algorithm 1,*

1. *Where is the* $\mathbf{b}$ *from? Why* $\mathbf{b}_0 = \mathbf{0}$*? Obviously, we want to get that* $\mathbf{b} \to \mathbf{0}$.
2. *What about* μ*?* $\mu \to \infty$ *or not ?*

Answer

1. $\boldsymbol{b}$ is the Bregman parameter, see [3,4] .
2. NOT, μ is fixed, here is SBI not classical ALM

Now, let's back to Eq 1.14, we get that

$$\min_{\alpha_G, \boldsymbol{u}} \frac{1}{2} \|H\boldsymbol{u} - \boldsymbol{y}\|_2^2 + \lambda \|\alpha_G\|_0, \quad s.t. \ \boldsymbol{u} = D_G \circ \alpha_G \tag{1.23}$$

Then, we define that

$$f(\boldsymbol{u}) = \frac{1}{2} \|H\boldsymbol{u} - \boldsymbol{y}\|_2^2, \quad g(\alpha_G) = \lambda \|\alpha_G\|_0 \tag{1.24}$$

Then, update $\boldsymbol{u}, \alpha_G$ *and* $\boldsymbol{b}$ by Eq 1.25

$$\begin{aligned} \boldsymbol{u}^{t+1} &= \arg\min_{\boldsymbol{u}} \frac{1}{2} \|H\boldsymbol{u} - \boldsymbol{y}\|_2^2 + \frac{\mu}{2} \left\|\boldsymbol{u} - D_G \circ \alpha_G^t - \boldsymbol{b}^t\right\|_2^2 \\ \alpha_G^{t+1} &= \arg\min_{\alpha_G} \lambda \|\alpha_G\|_0 + \frac{\mu}{2} \left\|\boldsymbol{u}^{t+1} - D_G \circ \alpha_G - \boldsymbol{b}^t\right\|_2^2 \\ \boldsymbol{b}^{t+1} &= \boldsymbol{b}^t - \left(\boldsymbol{u}^{t+1} - D_G \circ \alpha_G^{t+1}\right) \end{aligned} \tag{1.25}$$

$\boldsymbol{u}$ Sub-Problem

$$\min_{\boldsymbol{u}} Q_1(\boldsymbol{u}) = \min_{\boldsymbol{u}} \frac{1}{2}\|H\boldsymbol{u} - \boldsymbol{y}\|_2^2 + \frac{\mu}{2}\|\boldsymbol{u} - D_G \circ \alpha_G - \boldsymbol{b}\|_2^2 \tag{1.26}$$

By Eq 1.26 is convex and Fermat's Lemma, it is easy to get that

$$\boldsymbol{u}^* = \left(H^T H + \mu I\right)^{-1}\left[H^T\boldsymbol{y} + \mu\left(D_G \circ \alpha_G + \boldsymbol{b}\right)\right] \tag{1.27}$$

back to Eq 1.23 & Eq 1.13 .

Remark 1.4. *I do not check Eq 1.27 carefully.*

However, there are some drawbacks in inverse of a matrix.

Therefore, this paper [1] obtain the $\boldsymbol{u}$ via the gradient descent method as the following ,

$$\boldsymbol{u}^{t+1} = \boldsymbol{u}^t - \eta\left[H^T H\boldsymbol{u}^t - H^T\boldsymbol{y} + \mu\left(\boldsymbol{u}^t - D_G \circ \alpha_G - \boldsymbol{b}\right)\right] \tag{1.28}$$

When $\boldsymbol{u}^t \to \boldsymbol{u}^{t+1}$, we will get $\boldsymbol{u}^*$.

Remark 1.5. *What about η ?*

α_G Sub-Problem

Back to 2nd formula of Eq 1.25, we can get that

$$\min_{\alpha_G} Q_2(\alpha_G) = \min_{\alpha_G} \frac{1}{2}\|D_G \circ \alpha_G - \boldsymbol{r}\|_2^2 + \frac{\lambda}{\mu}\|\alpha_G\|_0 \tag{1.29}$$

where $\boldsymbol{r} = \boldsymbol{u} - \boldsymbol{b}$.

Let $\boldsymbol{x} = D_G \circ \alpha_G$, then Eq 1.29 can be write

$$\min_{\alpha_G} Q_2 = \min_{\alpha_G} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{r}\|_2^2 + \frac{\lambda}{\mu}\|\alpha_G\|_0 \tag{1.30}$$

In this paper [1], gives a key theorem

Theorem 1.1 ([1]). *Let $\boldsymbol{x}, \boldsymbol{r} \in \mathbb{R}^N, r_{G_k} \in \mathbb{R}^{\mathcal{B}_s \times c}$, and denote the error vector by $\mathbf{e} = \boldsymbol{x} - \boldsymbol{r}$, and $\mathbf{e}(j)$, where $j = 1, ..., N$.*

Assume that $\mathbf{e}(j)$ is independent and comes from a distribution with zero mean and variance σ^2. Then for any $\epsilon > 0$, we have the following property to describe the relationship between $\|\boldsymbol{x} - \boldsymbol{r}\|_2^2$ and $\sum_{k=1}^{n}\|\boldsymbol{x}_{G_k} - \boldsymbol{r}_{G_k}\|_F{}^2$

$$\lim_{\substack{N\to\infty \\ K\to\infty}} P\left\{\frac{1}{N}\|\boldsymbol{x} - \boldsymbol{r}\|_2^2 - \frac{1}{K}\sum_{k=1}^{n}\|\boldsymbol{x}_{G_k} - \boldsymbol{r}_{G_k}\|_F^2 < \varepsilon\right\} = 1 \tag{1.31}$$

By Thm 1.1, we can get that

$$\frac{1}{N}\left\|\boldsymbol{x}^t-\boldsymbol{r}^t\right\|_2^2=\frac{1}{K}\sum_{k=1}^{n}\left\|\boldsymbol{x}_{G_k}^t-\boldsymbol{r}_{G_k}^t\right\|_F^2 \tag{1.32}$$

with a large probability.

Eq 1.32 is equal to

$$\|\boldsymbol{x}-\boldsymbol{r}\|_2^2=\frac{N}{K}\sum_{k=1}^{n}\|\boldsymbol{x}_{G_k}-\boldsymbol{r}_{G_k}\|_F^2 \tag{1.33}$$

Back to 1.30

$$\begin{aligned}
&\arg\min_{\alpha_G}\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{r}\|_2^2+\frac{\lambda}{\mu}\|\alpha_G\|_0\\
=&\arg\min_{\alpha_G}\frac{N}{2K}\sum_{k=1}^{n}\|\boldsymbol{x}_{G_k}-\boldsymbol{r}_{G_k}\|_F^2+\frac{\lambda}{\mu}\|\alpha_G\|_0\\
=&\arg\min_{\alpha_G}\frac{1}{2}\sum_{k=1}^{n}\|\boldsymbol{x}_{G_k}-\boldsymbol{r}_{G_k}\|_F^2+\frac{\lambda K}{\mu N}\|\alpha_G\|_0\\
=&\arg\min_{\alpha_G}\frac{1}{2}\sum_{k=1}^{n}\|\boldsymbol{x}_{G_k}-\boldsymbol{r}_{G_k}\|_F^2+\frac{\lambda K}{\mu N}\sum_{k=1}^{n}\|\alpha_{G_k}\|_0\\
=&\arg\min_{\alpha_G}\sum_{k=1}^{n}\left(\frac{1}{2}\|x_{G_k}-r_{G_k}\|_F^2+\tau\|\alpha_{G_k}\|_0\right)
\end{aligned} \tag{1.34}$$

where $\tau=\frac{\lambda K}{\mu N}$.

Eq 1.34 can be efficiently minimized by solving n sub-problems for all group $\boldsymbol{x}_{G_k}$.

So,

$$\arg\min_{\alpha_{G_k}}\frac{1}{2}\|\boldsymbol{x}_{G_k}-\boldsymbol{r}_{G_k}\|_F^2+\tau\|\alpha_{G_k}\|_0=\arg\min_{\alpha_{G_k}}\frac{1}{2}\|D_{G_k}\alpha_{G_k}-r_{G_k}\|_F^2+\tau\|\alpha_{G_k}\|_0 \tag{1.35}$$

By Eq 1.17,1.19,1.20, $r_{G_k}=D_{G_k}\gamma_{G_k}$ where D_{G_k} is an unitary operator.

Back to Eq 1.35

$$\begin{aligned}
&\arg\min_{\alpha_{G_k}}\frac{1}{2}\|\boldsymbol{x}_{G_k}-\boldsymbol{r}_{G_k}\|_F^2+\tau\|\alpha_{G_k}\|_0\\
=&\arg\min_{\alpha_{G_k}}\frac{1}{2}\|D_{G_k}\alpha_{G_k}-r_{G_k}\|_F^2+\tau\|\alpha_{G_k}\|_0\\
=&\arg\min_{\alpha_{G_k}}\frac{1}{2}\left\|D_{G_k}\alpha_{G_k}-D_{G_k}\gamma_{r_{G_k}}\right\|_F^2+\tau\|\alpha_{G_k}\|_0\\
=&\arg\min_{\alpha_{G_k}}\frac{1}{2}\left\|\alpha_{G_k}-\gamma_{r_{G_k}}\right\|_F^2+\tau\|\alpha_{G_k}\|_0
\end{aligned} \tag{1.36}$$

Then Eq 1.36 can be solve by hard thresholding, see Section 1.5.

The solution of Eq 1.36 is $\alpha_{G_k}^* = hard\left(\gamma_{G_k}, \sqrt{2\tau}\right) = \gamma_{G_k} \odot 1\left(|\gamma_{G_k}| - \sqrt{2\tau}\right)$, where $\odot$ is element-wise product of two vector.

1.5 Hard Thresholding

The objective function is defined as follows:

$$f(x) = (x-b)^2 + \lambda |x|_0 \tag{1.37}$$

where $|x|_0 = \begin{cases} 0 & if\ x = 0 \\ 1 & if\ x \neq 0 \end{cases}$.

Then we can get that

$$f(x) = \begin{cases} b^2 & if\ x = 0 \\ (x-b)^2 + \lambda & if\ x \neq 0 \end{cases} \tag{1.38}$$

Whether $b^2 \geqslant \lambda$ or not?

Finally, we can get that

$$x^* = \arg\min_x f(x) = \begin{cases} 0, & f(x) = b^2, |b| \leqslant \sqrt{\lambda} \\ b, & f(x) = \lambda, |b| > \sqrt{\lambda} \end{cases} \tag{1.39}$$

Eq.1.39 is also called hard thresholding by the following formula

$$x^* = hard\left(b, \sqrt{\lambda}\right) \tag{1.40}$$

1.6 Acknowledge

We would like to thank Dr. Xu for distributing this paper to us. And we would also like to thank Dr. Zhang for help with describing the details in the GSR.

1.7 References

[1] J. Zhang, D. Zhao and W. Gao, "Group-Based Sparse Representation for Image Restoration," *IEEE Transactions on Image Processing* , 2014, Vol. 23(8), pp. 3336–3351.

[2] L. Bregman , "The Relaxation Method of Finding the Common Points of Convex Sets and Its Application to the Solution of Problems in Convex Programming."

[3] P. Getreuer , "Notes on Bregman Iteration," https://getreuer.info/posts/bregman.pdf.

[4] W. Yin , "The Bregman Methods: Review and New Results," https://www.caam.rice.edu/~optimization/L1/bregman/WotaoYin_Bregman_SIAMPDE_09.pdf.