

基于 BSSMF 矩阵分解的图像降噪方法

王伟钊

2025 年 3 月 30 日

摘要

目录

1 绪论	3
1.1 问题描述	3
1.2 国内外图像降噪算法研究现状	4
1.2.1 基于传统滤波器的方法	4
1.2.2 基于小波变换的方法	4
1.2.3 基于深度学习的方法	4
1.3 本文的研究内容	5
2 相关工作	6
2.1 BSSMF 算法	6
2.2 设定特定边界条件的 BSSMF 算法	8
2.3 对概率单纯形的改进	10
2.4 基于 BSSMF 的图像降噪算法	10
3 实验结果与分析	12
3.1 实验设计	12
4 结论	13
A 标准正态分布的 Z 分位数表	15

1 绪论

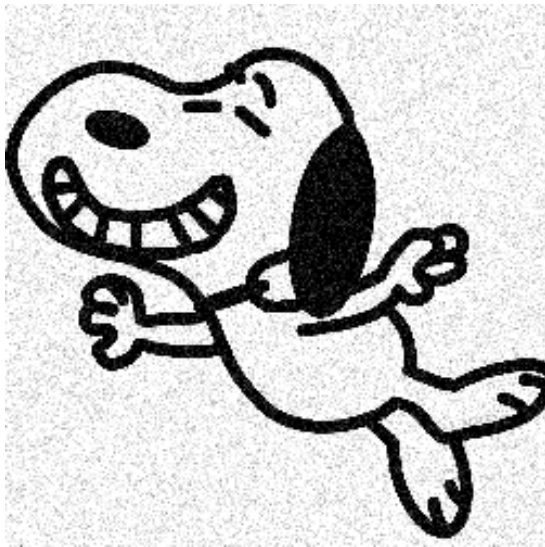
随着信息时代的快速发展，数字图像处理技术在各个领域得到了广泛的应用。然而，由于图像采集设备的限制，图像中常常会受到各种形式的噪声干扰。图像降噪是图像处理中的一个重要问题，其目的是去除图像中的噪声，使图像更加清晰，便于后续的图像分析和处理。图像降噪技术在医学图像处理、卫星图像处理、安防监控等领域有着广泛的应用。

1.1 问题描述

在数学上，图像降噪问题可以用如下数学表达式描述 [7]:

$$A(i, j) = A_0(i, j) + N(i, j) \quad (1)$$

其中 $A(i, j)$ 是观测到的图像， $A_0(i, j)$ 是原始图像， $N(i, j)$ 是噪声。噪声 $N(i, j)$ 可以是各种形式的，图1展示的是最常见的两种噪声：高斯白噪声和椒盐噪声。



(a) 高斯白噪声， $\mu=0$ ， $\sigma^2=30$



(b) 椒盐噪声

图 1: 高斯白噪声和椒盐噪声

本文主要关注高斯白噪声的降噪问题。高斯白噪声是一种均值为 0，方差为 σ^2 的高斯分布，其概率密度函数为：

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2)$$

其中 μ 是均值， σ 是标准差。由于大部分的白噪声均值为 0，因此估计 σ 成为了图像降噪中的一个关键步骤，因为它直接影响降噪算法的性能。准确估计噪声水平可以帮助选择合适的降噪参数，从而在保留图像细节和去除噪声之间取得平衡。如果 σ 估计不

准确，可能会导致过度平滑（损失细节）或降噪不足（残留噪声）[8]。为此，K. Rank 等人 [2] 使用绝对中位差（MAD）方法进行估计，Fabrizio Russo 等人 [9] 在 2003 年提出了一种基于新型滤波器对标准差 σ 进行估计的方法。Liu 等人在 2013 年提出了基于 SVD 的方法 [7] 进行估计。有了准确的噪声估计，我们就可以选择合适的降噪算法对图像进行降噪。

1.2 国内外图像降噪算法研究现状

随着人工智能的时代到来，图像降噪算法也经历了从传统方法到深度学习方法的转变。传统的图像降噪方法主要依赖于图像的先验知识和统计特性，而深度学习方法则通过大量的数据训练来自动学习图像的特征，从而实现更好的降噪效果。

1.2.1 基于传统滤波器的方法

传统的图像降噪方法主要包括均值滤波、中值滤波和高斯滤波等。这些方法通过对图像进行平滑处理来去除噪声，均值滤波是一种简单的线性滤波方法，通过对图像中每个像素点周围的像素值进行平均来平滑图像。中值滤波是一种非线性滤波方法，通过对图像中每个像素点周围的像素值进行排序，然后取中间值来平滑图像。高斯滤波是一种加权平均滤波方法，通过对图像中每个像素点周围的像素值进行加权平均来平滑图像。赵等人在 2011 年提出了一种能加速中值滤波算法计算的方法，使得中值滤波的效果更好以及计算更快 [14]。这些传统方法的底层逻辑十分简单，容易实现，但在处理复杂噪声时效果较差，且容易导致图像细节的损失，边缘模糊。

1.2.2 基于小波变换的方法

小波变换是一种时频分析方法，可以有效地提取图像的多尺度特征。基于小波变换的图像降噪方法通过对图像进行小波变换，将图像分解为不同频率的子带，然后对高频子带进行阈值处理，最后再进行小波逆变换来重构图像。徐等人在 2022 年提出了一种改进的小波软阈值函数 [12]，使得小波变换的去噪效果明显提升，这种方法可以有效地去除高频噪声，同时保留低频信息。但是，小波变换方法在处理图像边缘和细节时可能会出现伪影现象，导致图像质量下降。同时阈值函数的超参数的确定也会影响图片降噪效果。

1.2.3 基于深度学习的方法

近年来，深度学习在图像处理领域取得了显著的进展。基于深度学习的图像降噪方法主要包括卷积神经网络（CNN）和生成对抗网络（GAN）等 [15]，以及基于注意力机制的图像降噪方法 [13]。这些方法通过训练深度神经网络来学习图像的特征，从而实现图像降噪。与传统方法相比，基于深度学习的方法在处理复杂噪声和保留图像细节方面表现更好。但是，这些方法通常需要大量的训练数据和计算资源，且对网络结构和参数

的选择较为敏感。目前为止，仍然没有理论上的证明来说明深度学习方法的收敛性和稳定性。

1.3 本文的研究内容

本文主要研究基于有界单纯形矩阵分解 (Bounded Simplex Structured Matrix Factorization, BSSMF) 矩阵分解的图像降噪方法。BSSMF 算法是一种结合了非负矩阵分解 (NMF) 和结构化矩阵分解 (SSMF) 的矩阵分解算法。本文将 BSSMF 算法应用于图像降噪中，提出了一种新的图像降噪方法，并通过实验验证了其有效性。本文的主要贡献包括：

- 提出了一种基于 BSSMF 矩阵分解的图像降噪方法，通过对图像进行矩阵分解来去除噪声。
- 设计了一种新的 BSSMF 算法，通过引入特定的边界条件和概率单纯形约束来提高算法的性能。
- 通过实验验证了所提出的方法在不同噪声水平下的有效性，并与传统方法进行了对比分析。

2 相关工作

矩阵分解问题在数学上可以表示为：

$$X \approx WH \quad (3)$$

其中 $X \in \mathbb{R}^{m \times n}$ 是原始矩阵， $W \in \mathbb{R}^{m \times r}$ 和 $H \in \mathbb{R}^{r \times n}$ 是目标矩阵。矩阵分解要求 $r \ll \min(m, n)$ [3]，从而实现对原始矩阵的有效压缩和重构。后人们在此基础上加上了不同的约束条件，提出了多种矩阵分解算法，如非负矩阵分解 (NMF) [6]、稀疏矩阵分解 (SMF) [5]、结构化矩阵分解 (SSMF) [1] [11] 等。

2.1 BSSMF 算法

BSSMF (Bounded Sparse and Structured Matrix Factorization) [10] 是一种矩阵分解算法，旨在从给定的矩阵中提取出潜在的结构化信息。BSSMF 算法结合了 NMF 算法和 SSMF 算法，通过引入稀疏性约束和结构化约束来实现对矩阵的有效分解。该算法的主要思想是将(3)中的 W 和 H 矩阵进行约束，使得它们在分解过程中 W 满足边界条件(4)，

$$W(i, j) \in [a, b] \quad \forall i, j \quad (4)$$

而 H 矩阵满足单纯形约束(5)：

$$\Delta^r = \left\{ H \in \mathbb{R}^{r \times n} : H_{ij} \geq 0, \sum_{i=1}^r H_{ij} = 1, j = 1, 2, \dots, n \right\} \quad (5)$$

BSSMF 算法的目标是最小化以下目标函数：

$$f = \min_{W, H} \frac{1}{2} \|X - WH\|_F^2 \quad (6)$$

其中 λ_1 和 λ_2 是正则化参数， $\|\cdot\|_F$ 表示 Frobenius 范数， $\|\cdot\|_2$ 表示 L2 范数。BSSMF 算法通过迭代优化 W 和 H 矩阵，使得目标函数达到最小值。

迭代过程中，BSSMF 算法使用了梯度下降法，分别对目标函数(6)的 W 和 H 进行求梯度，得到以下更新公式：

$$\nabla f_W = -(X - WH) H^T \quad (7)$$

$$\nabla f_H = -W^T (X - WH) \quad (8)$$

BSSMF 算法通过迭代更新 W 和 H ，使得目标函数逐渐收敛到最小值。

以下是 BSSMF 算法1的计算过程：

Algorithm 1: BSSMF 算法

input : matrix $X \in \mathbb{R}^{m \times n}$, bounds $a \leq b \in \mathbb{R}$, initial factors $W \in \mathbb{R}^{m \times r}$ s.t.
 $W(:, k) \in [a, b]$ for all k and simplex structured $H \in \mathbb{R}_+^{r \times n}$, weights
 $M \in [0, 1]^{m \times n}$

output: W and H

- 1 $\alpha_1 = 1, \alpha_2 = 1, W_{old} = W, H_{old} = H,$
 $L_W^{prev} = L_W = \|HH^T\|_2, L_H^{prev} = L_H = \|WW^T\|_2$
- 2 **repeat**
- 3 **while** *stopping criteria not satisfied* **do**
- 4 $\alpha_0 = \alpha_1, \alpha_1 = (1 + \sqrt{1 + 4\alpha_1^2})/2$
- 5 $\beta_W = \min \left[(\alpha_0 - 1)/\alpha_1, 0.9999\sqrt{L_W^{prev}/L_W} \right]$
- 6 $\bar{W} \leftarrow W + \beta_W(W - W_{old})$
- 7 $W_{old} \leftarrow W$
- 8 $W \leftarrow \left[\bar{W} + \frac{M \odot (X - \bar{W}H)H^T}{L_W} \right]_b^a$
- 9 $L_W^{prev} = L_W$
- 10 $L_H \leftarrow \|W^T W\|_2$
- 11 **while** *stopping criteria not satisfied* **do**
- 12 $\alpha_0 = \alpha_2, \alpha_2 = (1 + \sqrt{1 + 4\alpha_2^2})/2$
- 13 $\beta_H = \min \left[(\alpha_0 - 1)/\alpha_2, 0.9999\sqrt{L_H^{prev}/L_H} \right]$
- 14 $\bar{H} \leftarrow H + \beta_H(H - H_{old})$
- 15 $H_{old} \leftarrow H$
- 16 $H \leftarrow \left[\bar{H} + \frac{W^T(M \odot (X - W\bar{H}))}{L_H} \right]_{\Delta^r}$
- 17 $L_H^{prev} = L_H$
- 18 $L_W = \|H^T H\|_2$
- 19 **until** *some stopping criteria is satisfied*;
- 20 注 *: L_W 和 L_H 分别表示 W 和 H 的 L2 范数, 可以有效控制算法速度, 且能在有部分数据为缺失的情况下进行收敛 [4]。

在原始的 BSSMF 算法中, 可以用 n 张具有相同特征的图片转为 n 个一维向量, 每张图片像素点的个数即为 m , 然后将 n 个向量拼接为 $m \times n$ 的输入矩阵 X , 算法运行后输入矩阵 X 的列可以近似表示为 W 矩阵中列的线性组合, 即:

$$X(:, j) \approx \sum_{i=1}^r W(:, i)H(i, j) \quad (9)$$

由于 H 矩阵满足单纯形约束, 因此 $H(i, j)$ 的值在 $[0, 1]$ 之间且加和为 1, 这更直观地说明了 X 的组成成分。在算法中对 W 进行 $[0, 255]$ 的约束, 可以使分解后的 W 直接对应一张图片, 更直观的解释了算法的可解释性。因此 BSSMF 算法通过对输入矩阵进

行分解，可以提取出潜在的特征信息，从而实现对数据的降维和特征选择。图2展示了 BSSMF 算法和 NMF 算法在 MNIST 数据集上的分解结果。可以看出，BSSMF 算法能够更清晰地提取出数字间共同的特征，而 NMF 算法则存在一定的模糊性。图3展示了数字 8 的 BSSMF 分解结果，以及最后的组合方式。可以看出，BSSMF 算法能够有效地提取出数字 8 的特征信息，并且通过对特征进行组合，可以重构出原始图像 [10]。

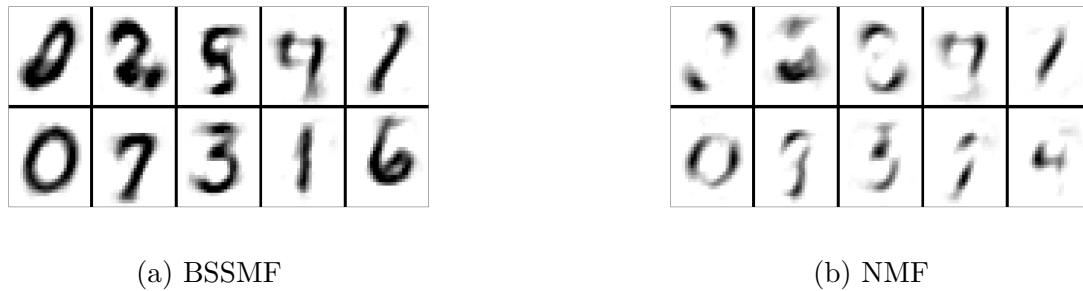


图 2: BSSMF 和 NMF 基于 500 张数字图像分解到 $r = 10$ 的对比

注 *: 每一张图片代表 W 矩阵中的一列经过 reshape 函数处理后的结果

$$\begin{array}{c}
 \boxed{\text{8}} = \boxed{\text{5}} + \boxed{\text{3}} + \boxed{\text{1}} + \dots \\
 \begin{array}{ccccccc}
 & 0.33\times & & 0.23\times & & 0.36\times & \\
 & \boxed{\text{5}} & + & \boxed{\text{3}} & + & \boxed{\text{1}} & + \dots
 \end{array}
 \end{array}$$

图 3: 数字 8 的 BSSMF 分解

2.2 设定特定边界条件的 BSSMF 算法

有了对 BSSMF 算法的初步了解后，我们可以对其进行改进。BSSMF 算法的一个缺点是分解后的 W 矩阵所有的列向量都在 $[a, b]$ 的范围内，这使得分解后的 W 矩阵的每一列都具有相同的边界条件，这在某些情况下可能不符合实际需求。例如，在图像降噪中，我们更希望分解后的 W 矩阵在前几列或第一列保留图像的最关键信息，而后面的列则只包含噪声信息 N 。为了解决这个问题，我们可以对 BSSMF 算法进行改进，设定特定的边界条件，使得分解后的 W 矩阵的每一列都具有不同的边界条件。我们可以将 W 矩阵的前 r_1 列设定为 $[a_1, b_1]$ ，后 r_2 列设定为 $[a_2, b_2]$ ，其中 $r_1 + r_2 = r$ 。这样，我们就可以在分解后的 W 矩阵中保留图像的最关键信息，同时去除噪声信息。如下图4所示，假设 W 矩阵的 W_{i1} 设定为 $[0, 255]$ ， W_{i2} 和 W_{i3} 设定为 $[0, 30]$ 。这样，我

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ x_{31} & x_{32} & x_{33} & x_{34} \\ x_{41} & x_{42} & x_{43} & x_{44} \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \end{bmatrix} \cdot \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \end{bmatrix}$$

图 4: 矩阵分解

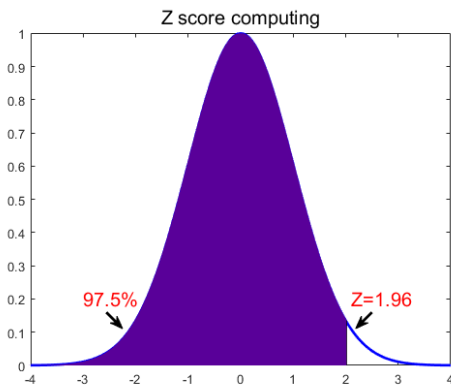
们在分解后的 W 矩阵中便能将图像的最关键信息保存在第 1 列，而噪声等其他信息保存在第 2 第 3 列。

除去对分解矩阵 W 第一列的约束为 $[0, 255]$ 外，我们应如何确定其他列的边界条件呢？基于高斯白噪声的概率学特性，可以找到绝大部分的噪声幅值所存在的范围。我们可以通过噪声的 Z 分位数来确定其他列的边界条件。

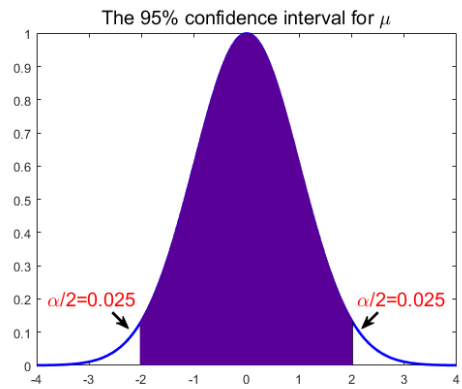
Z 分位数是一种用于描述数据分布的统计量，可以用于评估数据的集中趋势和离散程度。 Z 分位数的计算可用以下公式10表示：

$$Z = \frac{X - \mu}{\sigma} \quad (10)$$

其中 X 是数据值， μ 是均值， σ 是标准差。 Z 分位数所对应的概率值可以通过查找 Z 分位数表1计算获得。其内容为：在标准正态分布下 ($\mu = 0, \sigma = 1$)，随机变量 X 在 $P(X \leq z)$ 的概率。该表列出了不同分位数对应的 Z 值，可以用于快速查找和计算。但在计算的过程中需要注意的是， Z 分位数表中列出的概率值是单侧的，如图5a。而我们需要的是双侧的概率值 $P(X \leq z) + P(X \geq z)$ ，如图5b。因此，在查找 Z 分位数表时，需要将所需的概率值重新计算。



(a) 97.5% 的置信度



(b) 95% 的置信度

图 5: 标准正态分布

普通加性高斯白噪声的均值为 μ ，标准差为 σ ，我们可以通过查找 Z 分位数表1来确定噪声的幅值范围。假设我们希望分解能够包含 95% 的噪声，我们可以查找 Z 分位

数表中对应于 97.5% 的 Z 值为 1.96。那么 Z 在 $[-1.96, 1.96]$ 的概率对应为 95%。根据公式10，我们可以计算出噪声幅值范围为：

$$X \in [\mu - 1.96\sigma, \mu + 1.96\sigma] = [\mu - 1.96\sigma, \mu + 1.96\sigma]$$

因此，我们可以将 W 矩阵的后几列的边界条件设定为 $[\mu - 1.96\sigma, \mu + 1.96\sigma]$ 。

举一个例子，我们有一个高斯白噪声的图像，其均值为 10，标准差为 20。我们可以通过查找 Z 分位数表来确定噪声的幅值范围。我们希望分解能够包含 99% 的噪声，我们可以查找 Z 分位数表中对应于 99.5% 的 Z 值为 2.576。根据公式10，我们可以计算出噪声幅值范围为：

$$X \in [10 - 2.576 \times 20, 10 + 2.576 \times 20] = [-41.52, 61.52]$$

因此，我们可以将 W 矩阵的后几列的边界条件设定为 $[-41.52, 61.52]$ 。这样，我们就可以在分解后的 W 矩阵中保留图像的最关键信息，同时分离噪声信息。

2.3 对概率单纯形的改进

在原 BSSMF 算法中，输入矩阵 X 的列向量是随机打乱后输入的，这是由于打乱后的数据更能验证算法的鲁棒性。但若将 BSSMF 算法应用于图像降噪中，输入列向量所对应的标签可以作为一个重要的已知信息，合理运用这一信息可以提升算法的降噪效果。

由式9， H 的第 j 列直接对应于输入矩阵 X 的第 j 列。若已知 X 矩阵中第一列和第二列肯定包含图片信息，我们可以将 H 的第 j 列的第 i 行设定为 1，若不包含则设为 0。这样，我们就可以通过对 H 的第 1 行进行约束来确定原始图像中的主要成分。

现在举一个简单的例子。如图6所示，假设我们有一个包含两张图片以及两张噪声图片的输入矩阵 X ，其中第一列和第二列分别对应于两张图片与噪声的加和。我们可以将 H 的第 1 行设定为 $[1, 1, 0, 0]$ ，表示第一列和第二列包含图像信息，而第三第四张图片不含图片信息。这样，再分解的最后，我们在 W 矩阵中的第一列一定对应着我们图片的主要组成信息，而第 2 第 3 列则对应的噪声信息。

我们也可以通过设定 H 其他行列的值来实现对分解结果的约束。如第 1 第 3 行为 0，第 2 行为 1，这就约束了 W 的第二列为对应原始图像的主要组成信息。

2.4 基于 BSSMF 的图像降噪算法

下面基于上述对 BSSMF 算法的改进，提出一种新的基于 BSSMF 的图像降噪算法。首先为确保矩阵分解能够识别到图像中的重要信息，输入的图像应至少包含两张不同的含有图像细节以及随机的高斯噪声的图像。同时我们还需要至少一张空白的只含有高斯噪声的图像。将输入的图像通过 reshape 函数处理为一维向量后，拼接为一个 $m \times n$ 的矩阵 X ，其中 m 为每张图片像素点的个数， n 为图像数量。选择合适的 $r < \min(m, n)$ ，

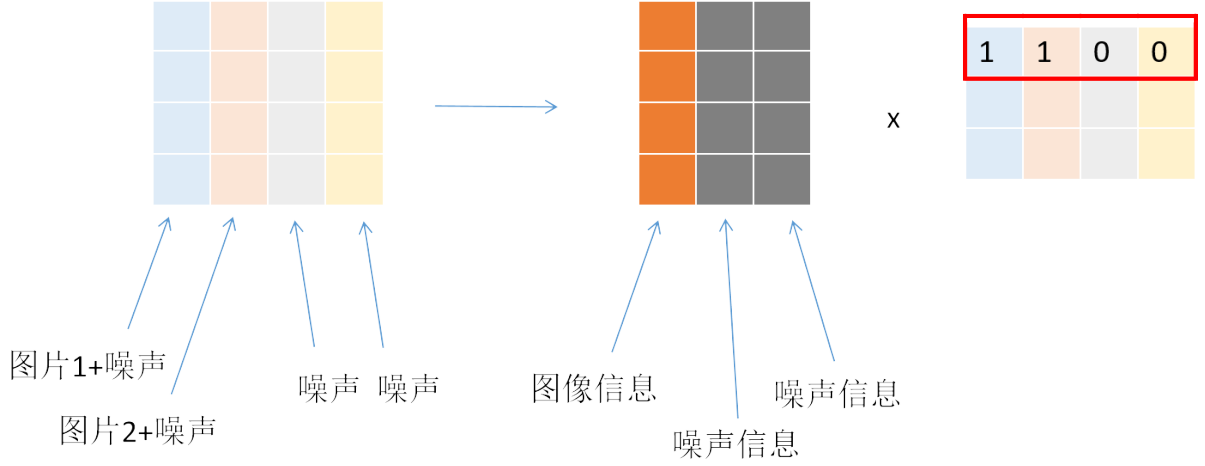


图 6: 对概率单纯形的改进

W 矩阵的第 1 列设定为 $[0, 255]$, 后 $r - 1$ 列设定为 $[\mu - 1.96\sigma, \mu + 1.96\sigma]$ 。然后将输入矩阵 X 输入到 BSSMF 算法中进行分解, 在每一次迭代后, 手动调整 H 第一行的值为 1 或 0, 直至完成计算。最后将分解后的 W 矩阵的第一列 reshape 为图片的格式, 得到降噪后的图像。下面给出改进的 BSSMF 算法2:

Algorithm 2: 改进的 BSSMF 降噪算法

input : matrix $X \in \mathbb{R}^{m \times n}$, bounds $a \leq b \in \mathbb{R}$, initial factors $W \in \mathbb{R}^{m \times r}$, image number $k < r$, noise parameter μ and σ , s.t.

$W(:, k) \in [a, b]$ for all k and simplex structured $H \in \mathbb{R}_+^{r \times n}$

output: W and H

```

1  $\alpha_1 = 1, \alpha_2 = 1, W_{old} = W, H_{old} = H,$ 
   $L_W^{prev} = L_W = \|HH^T\|_2, L_H^{prev} = L_H = \|WW^T\|_2$ 
2 repeat
3   while stopping criteria not satisfied do
4      $\bar{W} = \text{BSSMF\_update\_}W(W, H, X, L_H)$ 
5      $W(:, 1:k) \leftarrow \left[ \bar{W}(:, 1:k) \right]_b^a$ 
6      $W(:, k+1:r) = \left[ \bar{W}(:, k+1:r) \right]_{\mu+1.96\sigma}^{\mu-1.96\sigma}$ 
7      $L_H \leftarrow \|W^TW\|_2$ 
8     while stopping criteria not satisfied do
9        $H = \text{BSSMF\_update\_}H(H, W, X, L_W)$ 
10       $H \leftarrow [H]_{\Delta^r}$ 
11       $H(1, j) \leftarrow 1$  if  $j \leq k$  else 0
12       $L_W = \|H^TH\|_2$ 
13 until some stopping criteria is satisfied;

```

3 实验结果与分析

3.1 实验设计

4 结论

参考文献

- [1] Maryam Abdolali and Nicolas Gillis. Simplex-structured matrix factorization: Sparsity-based identifiability and provably correct algorithms. *SIAM Journal on Mathematics of Data Science*, 3(2):593–623, 2021.
- [2] David L Donoho and Iain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 09 1994.
- [3] Xiao Fu, Kejun Huang, Nicholas D. Sidiropoulos, and Wing-Kin Ma. Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications. *IEEE Signal Processing Magazine*, 36(2):59–80, 2019.
- [4] Le Thi Khanh Hien, Duy Nhat Phan, and Nicolas Gillis. An inertial block majorization minimization framework for nonsmooth nonconvex optimization, 2022.
- [5] Xiaobo Jin, Jianyu Miao, Qiufeng Wang, Guanggang Geng, and Kaizhu Huang. Sparse matrix factorization with $\ell_{2,1}$ norm for matrix completion. *Pattern Recognition*, 127:108655, 2022.
- [6] Daniel D. Lee and H. Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, Oct 1999.
- [7] Wei Liu and Weisi Lin. Additive white gaussian noise level estimation in svd domain for images. *IEEE Transactions on Image Processing*, 22(3):872–883, 2013.
- [8] K. Rank, M. Lendl, and R. Unbehauen. Estimation of image noise variance. *IEE Proceedings - Vision, Image and Signal Processing*, 146:80–84, 1999.
- [9] F. Russo. A method for estimation and filtering of gaussian noise in images. *IEEE Transactions on Instrumentation and Measurement*, 52(4):1148–1154, 2003.
- [10] Olivier Vu Thanh, Nicolas Gillis, and Fabian Lecron. Bounded simplex-structured matrix factorization: Algorithms, identifiability and applications. *IEEE Transactions on Signal Processing*, 71:2434–2447, 2023.
- [11] Ruiyuan Wu, Wing-Kin Ma, and Xiao Fu. A stochastic maximum-likelihood framework for simplex structured matrix factorization. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2557–2561, 2017.

- [12] 徐景秀, 张青. 改进小波软阈值函数在图像去噪中的研究应用. 计算机工程与科学, 44(01):92–101, 2022.
- [13] 王权. 一种注意力机制引导的双卷积神经网络图像去噪算法. *Computer Science and Application*, 15:159, 2025.
- [14] 赵高长, 张磊, 武风波. 改进的中值滤波算法在图像去噪中的应用. 应用光学, 32(4):678–782, 2011.
- [15] 刘迪, 贾金露, 赵玉卿, 钱育蓉. 基于深度学习的图像去噪方法研究综述. 计算机工程与应用, 57(7):1 – 13, 2021.

附录

A 标准正态分布的 Z 分位数表

表 1: 标准正态分布 Z 分位数表

Z 值	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817

注：该表展示了标准正态分布的 Z 分位数，Z 值表示标准正态分布曲线下的累积概率。