浙江大学

课程名称 矩阵论

任课教师 程磊

实验名称 图像去噪

姓　　名 赵一锟

学　　号 3220104685

年级专业 22 级 信息工程

所在学院 信息与电子工程学院

目录

1 实验目的

利用Tikhonov 正则化模型，将图像去噪问题描述为一个优化问题。结合课上学习到的优化知识，用梯度下降法对问题进行求解，获得去掉噪声后的图像。具体任务和示例如后面各小节所示，要求提交一份代码以及报告，说明各任务是如何实现的，以及自己的理解。

2 实验原理

2.1 数学模型

假设有一张带噪声的图片$Y \in \mathbb{R}^{m\times n}$ ，它和真实图片的关系为 $Y = X + E$，其中 $X \in \mathbb{R}^{m\times n}$是真实图片， 是高斯白噪声($E_{I,J} \sim N(0,\sigma^2)$)。图像去噪任务就是在给定带噪声图片Y的情况下，估计真实图片X（实际中真实图片是未知的，已知的只有带噪声的图片，这里给出真实图片是为了方便计算 PSNR 衡量去噪效果）。假设估计得到的真实图片表示为X_1。在Tikhonov 正则化模型中，X_1可以通过求解下面的优化问题得到：

$$X_1 = arg\,min\,\frac{1}{2}\|X - Y\|_F^2 + \lambda(\|D_1(X)\|_F^2 + \|D_2(X)\|_F^2)$$

其中 X 是优化变量，$D_1(X)$和$D_2(X)$分别表示 X 在水平和垂直方向上的向前差分，λ为正则系数。

函数形式说明：

•$\frac{1}{2}\|X - Y\|_F^2$：保持数据的一致性。它的作用是让去噪声后的图像 X 尽可能接近带有噪声的图像 Y，原因是去掉噪声后，图像的主要信息应与观测噪声图像保持一致。

•$\lambda(\|D_1(X)\|_F^2 + \|D_2(X)\|_F^2)$：正则化项抑制噪声。$D_1(X)$和$D_2(X)$分别表示 X 在水平和垂直方向上的向前差分，通过最小化这些差分的平方和们可以使得图像变得平滑，抑制图像中的高频噪声。参数λ控制正则化的强度。如果λ较大，模型会更倾向于平滑图像，去除噪声；如果λ较小，则更强调与观测图像的相似性。

•因此，该损失函数通过平衡数据一致性和图像平滑性来实现图像去噪。

在图像处理中，差分算子可以理解为一阶导数的离散化，而差分的导数可以理解为二阶导数的离散化，用拉普拉斯算子来描述：

$$[L(X)]_{i,j} = X_{i+1,j} + X_{i-1,j} + X_{i,j+1} + X_{i,j-1} - 4X_{i,j}$$

于是 f(X)的导数可以表示为：

$$\frac{df(X)}{dX} = X - Y - 2\lambda L(X)$$

2.2 BB 步长

考虑梯度下降法的格式：

$$x^{k+1} = x^k - D^k \nabla f(x^k)$$
$$D^k = \alpha_k I$$

BB 方法选取的α_k是如下两个最优问题之一的解：

$$\min\left\|\alpha y^{k-1} - s^{k-1}\right\|^2$$

$$\min\left\|y^{k-1} - \alpha^{-1} s^{k-1}\right\|^2$$

$$s^{k-1} = x^k - x^{k-1}$$
$$y^{k-1} = \nabla f(x^k) - \nabla f(x^{k-1})$$

可以解得上述最优问题的解为：

$$\alpha_{BB1}^{k} = \frac{(s^{k-1})^T y^{k-1}}{(y^{k-1})^T y^{k-1}}$$
$$\alpha_{BB2}^{k} = \frac{(s^{k-1})^T s^{k-1}}{(s^{k-1})^T y^{k-1}}$$

因此 BB 方法的两种迭代形式为：

$$x^{k+1} = x^k - \alpha_{BB1}^{k} \nabla f(x^k)$$
$$x^{k+1} = x^k - \alpha_{BB2}^{k} \nabla f(x^k)$$

算法 6.2 非单调线搜索的 BB 方法

给定 x^0，选取初值 $\alpha > 0$，整数 $M \geqslant 0$，$c_1, \beta, \varepsilon \in (0,1)$，$k = 0$.
while $\|\nabla f(x^k)\| > \varepsilon$ **do**
  while $f(x^k - \alpha \nabla f(x^k)) \geqslant \max\limits_{0 \leqslant j \leqslant \min(k,M)} f(x^{k-j}) - c_1 \alpha \|\nabla f(x^k)\|^2$ **do**
    令 $\alpha \leftarrow \beta\alpha$.
  end while
  令 $x^{k+1} = x^k - \alpha \nabla f(x^k)$.
  根据公式(6.2.9)之一计算 α，并做截断使得 $\alpha \in [\alpha_m, \alpha_M]$.
  $k \leftarrow k + 1$.
end while

2.3 核范数

核范数是一种用于矩阵正则化的方法，常用于处理低秩矩阵恢复等问题。在许多实际问题中，我们希望找到一个低秩矩阵的近似解，例如图像去噪。核范数的正则化通常用于鼓励矩阵的低秩结构，低秩矩阵能更好地反映数据的内在结构和隐藏模式。直接使用矩阵的秩约束问题是一个 NP-hard 问题，核范数提供了一种解决此问题的凸松弛方法，因为核范数是秩函数的凸包逼近。

核范数倾向于将小奇异值压缩到接近零，从而使得矩阵中的冗余信息被去除，得到近似低秩的矩阵表示。这种效果在图像处理、信号处理等需要提取主要特征的场景中尤为显著。在矩阵补全问题（如推荐系统）中，核范数正则化能在数据不完备的情况下有效地恢复矩阵的低秩结构。

给定一个矩阵 $X \in \mathbb{R}^{m \times n}$，其核范数定义为所有奇异值的和，即：

$$\|X\|_* = \Sigma_{i=1}^{min(m,n)} \sigma_i$$

核范数的正则化模型：

$$\min_X \frac{1}{2} \sum_{(i,j)\in\Omega} (X_{ij} - Y_{ij})^2 + \lambda \|X\|_*$$

核范数的梯度计算：假设 $X = U\Sigma V^T$，则核范数的此梯度为：

$$\partial \|X\|_* = UV^T$$

3 实验内容-基础部分

3.1 加载图像，并进行图像预处理

使用 matlab 中的 imread、rgb2gray、imshow 函数加载灰度图像，并未图像添加$\sigma^2 = 20$的高斯白噪声。

图 1 图像预处理

3.2 计算函数值与梯度

根据上述分析的数学建模中计算函数值与梯度的公式，用 matlab 具体实现如下：

```
function [value, grad] = gradient_value(x, y, lambda)
%函数用于计算函数值和梯度，用于后面的梯度下降算法
%输入：x，y，lambda（正则系数）
%输出：函数值 value，梯度 grad
    %差分矩阵计算
    D1x = x(:, [2:end, end]) - x;
    D2x = x([2:end, end], :) - x;

    %函数值计算
    value = 0.5 * norm(x - y, 'fro')^2 + lambda * (norm(D1x, 'fro')^2 + norm(D2x, 'fro')^2);

    %梯度计算
    L_x = x([2:end, end], :) + x([1, 1:end-1], :) + x(:, [2:end, end]) + x(:, [1, 1:end-1]) - 4 * x;
    grad = x - y - 2 * lambda * L_x;
end
```

通过上述过程，我们可以计算函数当前的函数值与梯度值。

3.3 梯度下降（constant 步长）

3.3.1 梯度下降基本格式

$$x^{k+1} = x^k - \mu_k \nabla f(x^k)$$
$$\mu_k = constant$$

具体 matlab 实现代码如下：

```
for iter = 1:iters
        %计算函数值及梯度
        [value, grad] = gradient_value(x, y, lambda);
        %更新 x
        x = x - step * grad;
```

3.3.2 不同正则化系数下的降噪结果比较

图 2 step=0.1 固定步长降噪

可以观察到，随着 λ 的增大，降噪后的图片越来越平滑，与理论情况相符。

3.3.3 梯度与函数值收敛性分析

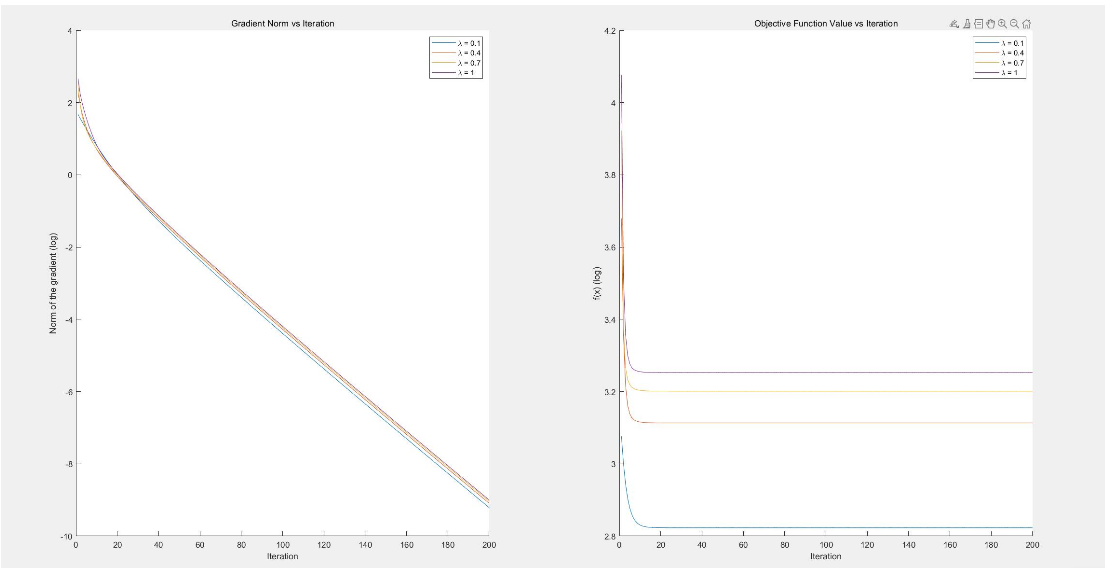

图 3 constant 收敛曲线（step = 0.1）

我们可以看到，所有不同的 λ 值对应的梯度范数都在逐渐减小，并且在 200 次迭代后趋近收敛。不同的 λ 值对收敛速度的影响相对较小。无论 λ 的取值是 0.1、0.4、0.7 还是 1，梯度范数的下降曲线几乎是相似的。

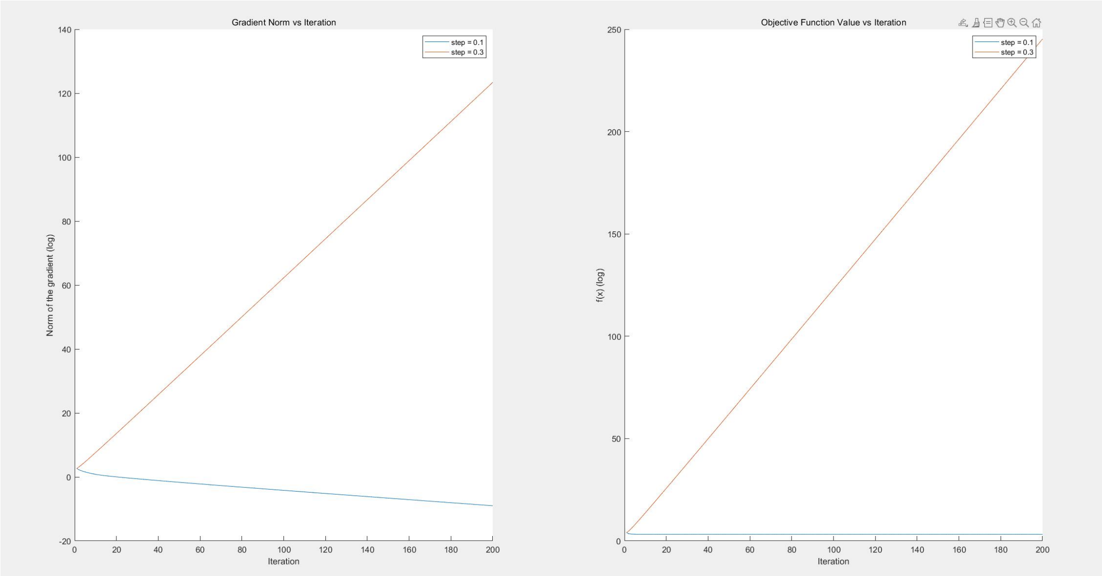

图 4 不同 step 收敛曲线

如上图所示，当 step = 0.3 时，梯度下降算法不收敛。因此，步长对收敛速度起到的影响更大。

3.3.4 psnr 值分析

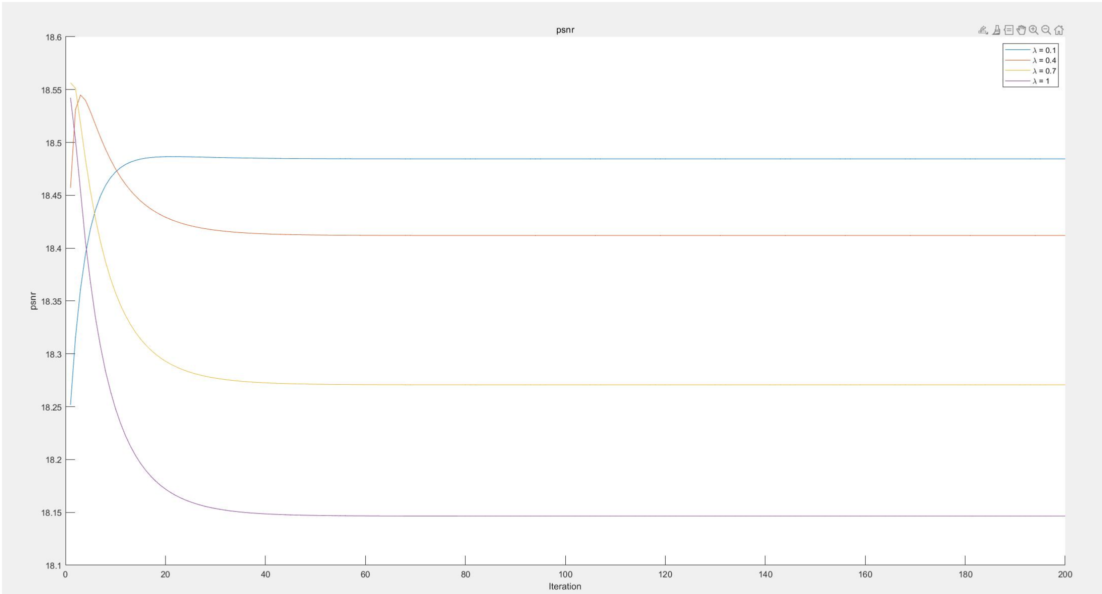

图 5 psnr 值（step=0.1）

psnr 值分析中，λ 越小，psnr 值相对越高，这与观测到的降噪平滑程度效果图略有不同。

4 实验内容-选做部分

4.1 BB 步长相关分析

结合在实验原理部分的分析，我们对梯度下降算法作出以下修改，采用 BB 步长进行优化：

```
%BB 步长
if iter > 1
    s = x - x_prev;
    y_diff = grad - grad_prev;
    step = (s(:)' * s(:)) / (s(:)' * y_diff(:) + eps);
end

%更新 x
x_prev = x;
grad_prev = grad;
x = x - step * grad;
```

相比于 constant 步长，BB 步长的主要优势在于算法的收敛速度快，可节约大量时间。

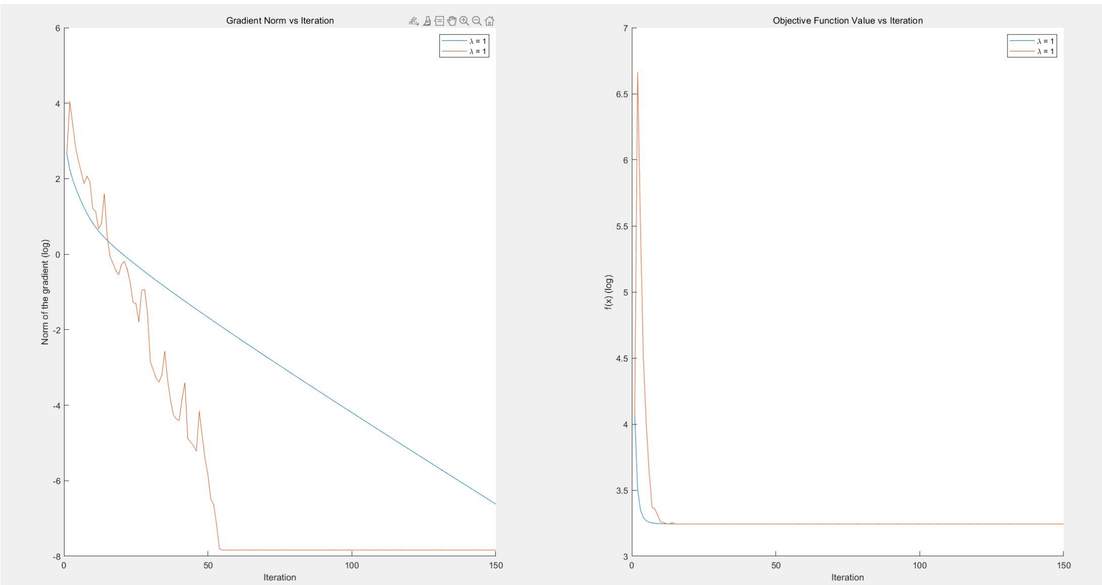

图 6 不同步长方法的收敛性

通过比较可以看到，BB 步长在 50—60 次迭代后梯度下降基本完成，而 constant 步长仍在继续。

4.2 图像存在缺失值

在有噪声的情况下，假设图像存在缺失值，即智能观测到部分有噪声的数据。图像的优化问题即可以建模为：

$$\arg\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{Y} - \mathbf{O} * \mathbf{X}\|_F^2 + \lambda\phi(\mathbf{X})$$

使用原来的梯度下降法，降噪后的结果为全噪声图像，这是由于图像存在缺失值造成的数据缺失问题，因此我们令 ϕ (X)为核范数来优化正则化模型，使其能够对低秩矩阵有更好的逼近。对损失函数的函数值与梯度值计算方法进行更新：

```
% 核范数梯度计算
```

```
    [U, S, V] = svd(x, 'econ');
    S_threshold = max(S - lambda, 0); % 奇异值软阈值化
    x_nuclear = U * S_threshold * V';

    value = 0.5 * norm(O .* (y - x), 'fro')^2 + lambda * sum(diag(S)); % 损失函数值的计算

    % 梯度计算
    grad = O .* (x - y) + (x - x_nuclear);
```

接下来，利用 BB 步长进行梯度下降，再将得到的图像使用原梯度下降算法进行梯度下降，得到最终降噪的结果，如下图：

图 7 存在缺失时的降噪结果

根据初步的降噪结果，可以看到降噪后的图像相比原图像略显“朦胧”。接下来我们分析两次降噪的 psnr

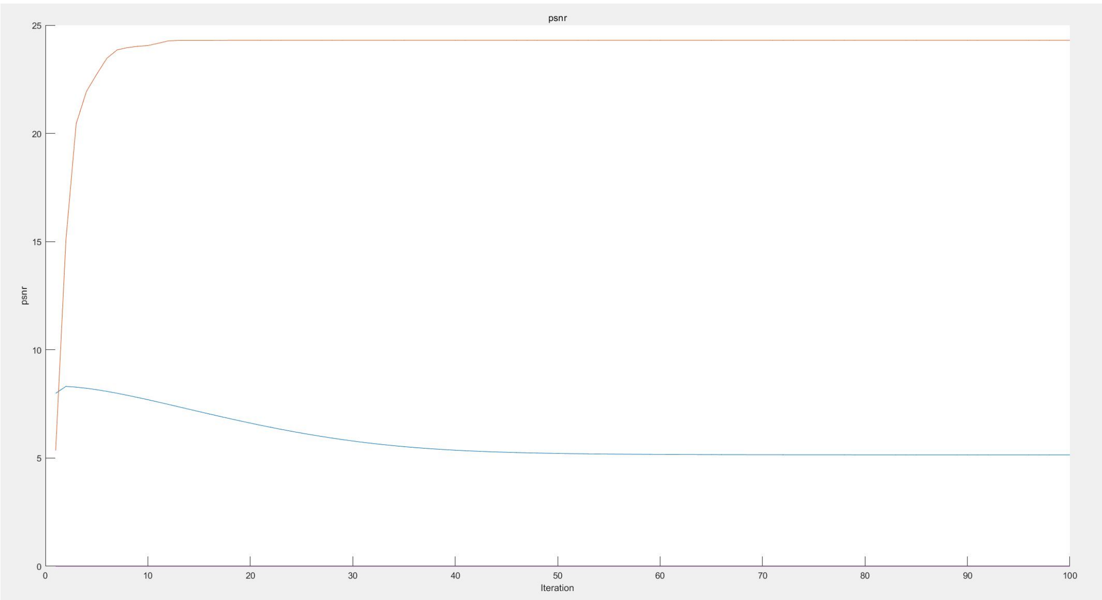

图 8 存在缺失降噪 psnr

蓝色曲线为使用核范数进行逼近的 psnr，此时 psnr 仍然较小，但其解决了低秩矩阵求解的问题，橙色曲线为在核范数正则化的基础上，利用原梯度下降法进行降噪后的 psnr，可以看到此时 psnr 的值上升到了 25 左右。两次结合运用，降噪效果较好。

5 总结与反思

在这次图像降噪大作业的学习中，我学会了一些基础的图像降噪方法，运用了课上所学习的 Tikhonov 正则化模型、奇异值分解、梯度下降等等理论知识。在实际的工程运用中，这些看似枯燥的公式在图像处理的过程中也展现出了灵动的趣味，令人回味无穷。

美中不足的是，选做部分处理信息缺失的图像降噪问题的结果仍有很大的提升空间。与此同时，EVD 分解的时间复杂度为立方级别，使用 matlab 进行梯度下降的迭代过程中耗时较长。

矩阵论的课堂是大学中难能可贵的教学时刻，感谢程磊老师细致的讲解与指导，也很幸运遇见一群志同道合的人共勉。

I love matrix.