课程作业 (一)

李子牛，沈炜霖，陈辉

2018 年 11 月 25 日

1 问题

课程作业要求我们求解如下问题：

$$\begin{aligned} &\min ||\nabla u||_p \\ &\text{subject to } Au = b \end{aligned} \tag{1}$$

其中等式约束由随机的傅里叶采样决定。实验中，要求给出在 10 条线的采样下，$p = 0.1, 0.5, 0.8, 1$ 时的 Phantom 恢复图像。

2 算法

在求解问题时，我们参考 [1] 提出的投影梯度法。具体而言，输入平滑参数 ϵ, 衰减因子 α，和迭代精度 δ，初值值 u_0；每一步，算法先根据目标函数计算出梯度，然后搜索最优的步长以满足等式约束，最终迭代输出恢复的图像 u_n。算法框架见1。

3 结果

实验中，我们采用 $\epsilon = 10, \alpha = 10$, 初始值为 mini-energy 恢复的结果，最大迭代次数 2000，采样矩阵参考 l1magic 的代码。

3.1 10 条线

按照题目要求，我们给出 10 条线采样下的结果，见图1、2、3、4。虽然恢复的效果很差，但可以看出，相同的采样下，p 越大，恢复的效果越好。

Algorithm 1 投影梯度法

H

repeat

$d_n = -\mathrm{div}\Big((\sqrt{\nabla u_n{}^2 + \epsilon^2})^{p-2}\nabla u_n\Big)$

if $\mathrm{mod}(n, 100) == 0$ **then**

step $= \min_t ||A(u_n + td_n) - b||_2$

end if

if $\mathrm{mod}(n, 300) == 0$ **then**

$\epsilon := \epsilon/\alpha$

end if

$u_{n+1} = u_n + \mathrm{step} * d_n$

until $||u_n - x||_\infty < \delta$

图 1: p=0.1, 10 条线

图 2: p=0.5, 10 条线

图 3: p=0.8, 10 条线

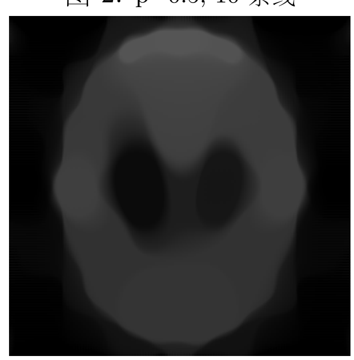

图 4: p=1.0, 10 条线

3.2 30 条线

为了验证我们的算法确实在极小化 TV 的 l_p-norm, 我们增了采样数，为 30 条线，见图5、6、7、8。可以看到效果相比 10 条线的采样，有很大的提升。

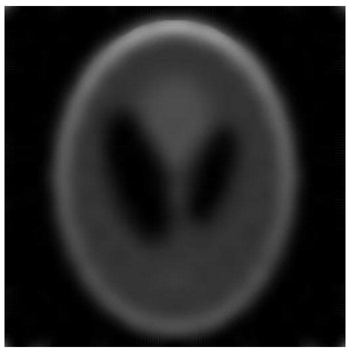

图 5: p=0.1, 30 条线

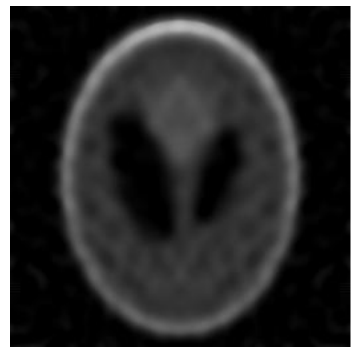

图 6: p=0.5, 30 条线

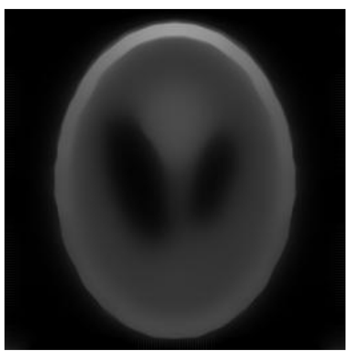

图 7: p=0.8, 30 条线

图 8: p=1.0, 30 条线

4 分析

实验中，正如算法所示，加入 ϵ 会平滑，以及使得 $(\sqrt{\nabla u^2})^{p-2}$ 有意义，但是算法的初始阶段, ϵ 较大，得到的值几乎为 0，算法几乎不下降；在 ϵ 衰减几次后，其不起作用，极小化梯度会使得图像尽可能的平滑 (去噪声)。但是，我们没能很好地控制 ϵ，造成图像的过度平滑。

5 代码

```
clear;clc;
path(path, './Optimization');
path(path, './Measurements');
path(path, './Data');

% Phantom
n = 256;
N = n*n;
X = phantom(n);
x = X(:);
% l-p norm
epsilon = 10;
factor = 10;
p = 1.0;

% number of radial lines in the Fourier domain
L = 30;

% Fourier samples we are given
[M,Mh,mh,mhi] = LineMask(L,n);
OMEGA = mhi;
A = @(z) A_fhp(z, OMEGA);
At = @(z) At_fhp(z, OMEGA, n);

% measurements
y = A(x);

% min l2 reconstruction (backprojection)
xbp = At(y);
Xbp = reshape(xbp,n,n);

% recovery
tic
[gx, gy] = gradient(X);
tvI = sum(sum(sqrt(gx.^2 + gy.^2)));
fprintf('Original TV = %8.3f\n', tvI);

un = Xbp;
i = 0;
step = 0.01;
dis = 1000;
while true
    i = i + 1;
    [gx, gy] = gradient(un);
```

```
    p_norm = sum(sum(power(sqrt(gx.^2 + gy.^2 ), p)));

    coef = power(sqrt(gx.^2 + gy.^2 + epsilon^2), p-2);
    tvU_x = coef .* gx;
    tvU_y = coef .* gy;

    dn = -divergence(tvU_x, tvU_y);
    j = 0;
    if mod(i, 100) == 0
        for t = 1e-1: -1e-3 : 1e-4
            j = j + 1;
            u_next = un - t*dn;
            y_next = A(u_next(:));
            if abs(norm(y_next) - norm(y)) < sqrt(epsilon)/100
                break;
            end
        end
        step = t;
    end
    un = un - step*dn;
    [gx, gy] = gradient(un);
    p_norm_next = sum(sum(power(sqrt(gx.^2 + gy.^2 ), p)));
    tol = max(max(un-X));
    if max(max(un-X)) < dis
        dis = tol;
        imwrite(un,['figure/best.jpg']);
    end
    if mod(i, 100) == 0
        fprintf('i=%i, j=%i, rate=%.3f \n', i, j, p_norm_next/p_norm);
    end
    if p_norm_next < p_norm * 0.95 || mod(i, 300) == 0
        fprintf('shrink \n');
        epsilon = epsilon / factor;
        imwrite(un,['figure/', int2str(i), '.jpg']);
    end
    if i > 2000
        break;
    end
end
toc
Xtv = un;
figure();
subplot(1, 3, 1);
imshow(X);
subplot(1, 3, 2);
imshow(Xbp);
```

```
subplot(1, 3, 3);
imshow(Xtv);
```

homwork.m

参考文献

[1] Rick Chartrand, "Exact reconstructions of sparse signals via nonconvex minimization," IEEE Signal Processing Letters, vol. 14, pp. 707–710, 2007