问题： $\min f(x) = \min g(x) + h(x)$

其中 $g(x)$ 凸 可微（光滑）

$h(x)$ 凸 部分不可微

对于 $h(x)$ 找一个代理，记作 近端算子 proximal operator

$$\mathrm{prox}_{h,t}(x) = \underset{z}{\mathrm{argmin}}\ \frac{1}{2t}\|x-z\|_2^2 + h(z) \quad ①$$

x、z 要接近　　　　组作为 $g(x)$ 的代理

① 和 $g(\cdot)$、x 无关，∴ 可先求凸且可微的 $g(x)$ 的 min $\hat{x}$，

再代入 ① 中求 z，即可得整个目标函数 $f(x)$ 的 min，

这是一个两步迭代过程。

∴ PGD 过程如下：

$$\hat{x}^{k+1} = x^k - t\nabla g(x^k)$$

$$x^{k+1} = \mathrm{prox}_{h,t}(\hat{x}^{k+1}).$$

即 $x^{k+1} = \mathrm{prox}_{h,t}(x^k - t\nabla g(x^k))$ ②

下面是 ② 成立的证明

$$x^{k+1} = \mathrm{prox}_{t,h}(x - \nabla g(x^k))$$

$$= \arg\min_z h(z) + \frac{1}{2t}\| z - (x^k - t\nabla g(x^k))\|_2^2$$

$$= \arg\min_z h(z) + \frac{t}{2}\|\nabla g(x^k)\|_2^2 + \nabla g(x)^T(z - x^k) + \frac{1}{2t}\| z - x^k\|_2^2$$

与(z)无关的常数

$$= \arg\min_z h(z) + g(x^k) + \nabla g(x)^T(z - x^k) + \frac{1}{2t}\| z - x^k\|_2^2$$

近似 $g(z)$ 在 x^k 处的二阶泰勒展开

$$\doteq \arg\min h(z) + g(z).$$

一些例子.

① $h(x)=0$ 时.

$$\mathrm{prox}_{h,t} = \arg\min_z \frac{1}{2t}\| x - z\|_2^2 + 0 = x.$$

$$x^{k+1} = x^k - t_k \nabla g(x^k).$$

退化为梯度下降.

② $h(x) = I_C(x)$ 指示函数

$$\operatorname{prox}_{t,h}(x) = \arg\min_z \frac{1}{2t}\|x-z\|_2^2 + I_C(z)$$

$$= \arg\min_{z\in C} \frac{1}{2t}\|x-z\|_2^2$$

$$= P_C(x)$$

$$x^{k+1} = P_C(x^k - t_k \nabla g(x^k))$$

③ $h(x) = \|x\|_1$ 软阈值算法

$$\operatorname{prox}_{t,h} = \arg\min_z \frac{1}{2t}\|x-z\|_2^2 + \|z\|_1$$

$$= \arg\min_z \underbrace{\|x-z\|_2^2 + 2t\|z\|_1}$$

$$= S_t(x)$$

$H(z)$ 和 x 无关

$H'(z) = -2(x-z) + 2t\|\ \|'$

$\quad \| \quad 0$

$\Rightarrow x - z = t\|\ \|'$

$z = x - t\|\ \|'$

$= x - \operatorname{sign}(z)$

$$S_t(x) = \begin{cases} x-t & x > t \\ 0 & |x| \le t \\ x+t & x < -t \end{cases}$$

近端梯度算法（Proximal Gradient Method）

问题：$\min f(x) = \min g(x) + h(x)$

其中 $g(x)$ 凸 可微（光滑）

$h(x)$ 凸 部分不可微

对于 $h(x)$ 找一个代理 记作 近端算子 proximal operator

$$\mathrm{prox}_{h,t}(x) = \underset{z}{\arg\min}\ \underbrace{\frac{1}{2t}\|x-z\|_2^2}_{x\ z\text{要接近}} + \underbrace{h(z)}_{\text{留作为}g(x)\text{之代理}} \quad ①$$

① 和 $g(\cdot)$、x 无关，∴ 可先求凸且可微的 $g(x)$ 之 $\underline{\min}$ 值.

PGD.pdf

```
function [x]=proximalGradient(A,b,gamma)
%%解决  1/2||AX-b||2_{2}+gamma*||X||1
%% A:m*n    X:n*1

tic;
MAX_ITER =400;
ABSTOL   = 1e-4;
RELTOL   = 1e-2;

f = @(u) 0.5*norm(A*u-b)^2;%%为了确定线搜索步长
lambda = 1;  % 步长
beta = 0.5;
[~,n]=size(A);
x = zeros(n,1);
xprev = x;
AtA = A'*A;
Atb = A'*b;
for k = 1:MAX_ITER
%     while 1
```

```
        grad_x = AtA*x - Atb;
        % gamma表示软阈值函数的阈值（非负值）
        % z是要更新的递推迭代式的xk
        z = soft_threshold(x - lambda*grad_x, lambda*gamma);%%迭代更新x
        %线性搜索步长（注释掉即为固定步长）
%         if f(z) <= f(x) + grad_x'*(z - x) + (1/(2*lambda))*(norm(z - x))^2
%             break;
%         end
%         lambda = beta*lambda;   %减小步长

    xprev = x;
    x = z;

    h.prox_optval(k) = objective(A, b, gamma, x, x);
    if k > 1 && abs(h.prox_optval(k) - h.prox_optval(k-1)) < ABSTOL
        break;
    end
end
% %==========================================
% % 得到目标函数的解x以及x所对应的函数近似值p_prox
% %==========================================
h.x_prox = x; % 得到的近端解x
h.p_prox = h.prox_optval(end); % 函数的近端近似值?
% %==========================================
% % 显示程序运行的相关信息
% %==========================================
h.prox_grad_toc = toc; % 计时程序运行时间
fprintf('Proximal gradient time elapsed: %.2f seconds.\n', h.prox_grad_toc);
h.prox_iter = length(h.prox_optval);
K = h.prox_iter;
h.prox_optval = padarray(h.prox_optval', K-h.prox_iter, h.p_prox, 'post');

plot( 1:K, h.prox_optval, 'r-');
xlim([0 75]);


end

function p = objective(A, b, gamma, x, z)
%目标函数的代码形式
  p = 0.5*(norm(A*x - b))^2 + gamma*norm(z,1);
end

function [X]=soft_threshold(b,lambda)
% 软阈值函数
  X=sign(b).*max(abs(b) - lambda,0);
end
```

Ref:

对近端梯度算法(Proximal Gradient Method)的理解_chaolei_9527的博客-CSDN博客_prox函数

优化算法之近端梯度下降（proximal gradient descent）

机器学习 | 近端梯度下降法 (proximal gradient descent)

近端梯度下降算法(Proximal Gradient Algorithm)_Ten_yn的博客-CSDN博客_近端梯度下降