Sample 12-4

画像復元

高速繰り返し縮退閾値アルゴリズム (FISTA)

画像処理特論

村松 正吾

動作確認: MATLAB R2020a

Image restoration

Fast iterative shrinkage/thresholding algorithm (FISTA)

Advanced Topics in Image Processing

Shogo MURAMATSU

Verified: MATLAB R2020a

準備

(Preparation)

```
clear
close all
import msip.download_img
msip.download_img
```

```
lena.png already exists in ./data/
baboon.png already exists in ./data/
goldhill.png already exists in ./data/
barbara.png already exists in ./data/
```

問題設定

(Problem settings)

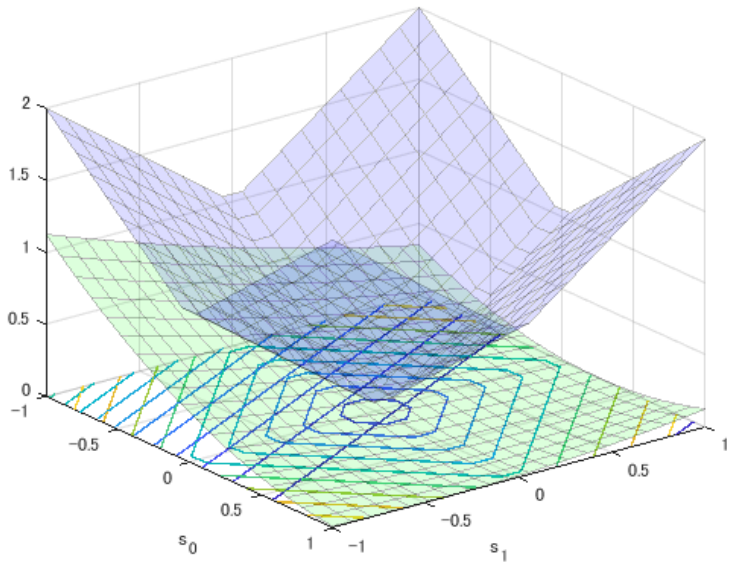
$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{v} - \mathbf{D}\mathbf{s}\|_2^2 + \lambda\|\mathbf{s}\|_1$$

- $\mathbf{D} = \begin{pmatrix} \frac{2}{3} & \frac{1}{3} \end{pmatrix}: \quad \mathbb{R}^2 \to \mathbb{R}^1$
- $\mathbf{v} = \frac{1}{2} \in \mathbb{R}^1$
- $\lambda \in [0, \infty)$
- $\mathbf{s} \in \mathbb{R}^2$

```
D = [2 1]/3;
v = 0.5;
```

関数プロット
(Function plot)

```
% Function settings
f = @(s0,s1) 0.5*(v-(D(1)*s0+D(2)*s1)).^2;
g = @(s0,s1) (abs(s0)+abs(s1));
% Variable settins
s0 = linspace(-1,1,20);
s1 = linspace(-1,1,20);
[S0,S1] = ndgrid(s0,s1);
F = f(S0,S1);
G = g(S0,S1);
% Surfc plot of the fidelity
figure(1)
hf = surfc(s0,s1,F);
hf(1).FaceAlpha = 0.125;
hf(1).FaceColor = 'green';
hf(1).EdgeAlpha = 0.25;
hf(2).LineWidth = 1;
set(gca,'YDir','reverse');
hold on
% Surfc plot of the regularizer
hg = surfc(s0,s1,G);
hg(1).FaceAlpha = 0.125;
hg(1).FaceColor = 'blue';
hg(1).EdgeAlpha = 0.25;
hg(2).LineWidth = 1;
xlabel('s_1')
ylabel('s_0')
hold off
```



パラメータ設定
(Parameter settings)

```
lambda = 0.2;
gamma = 0.4;
niters = 20;
```

ℓ_1 -ノルム正則化最小自乗法による近似
(ℓ_1 -norm-regularized least square method)

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \frac{1}{2}\|\mathbf{v} - \mathbf{D}\mathbf{s}\|_2^2 + \lambda\|\mathbf{s}\|_1$$

近接勾配法に帰着させる．(Reduced to a proximal gradient method)

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in V} f(\mathbf{x}) + g(\mathbf{x})$$

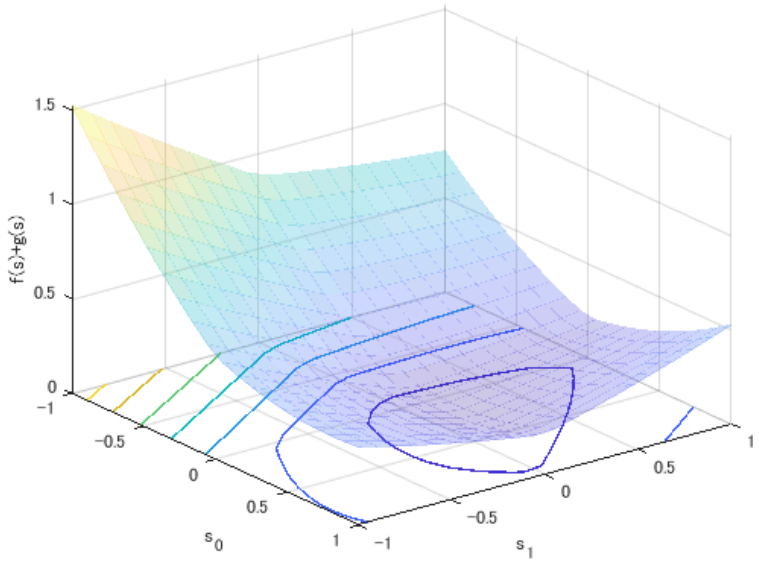
- $f(\cdot), g(\cdot) \in \Gamma_0(\mathbb{R}^L)$: Convex functions
- $f(\cdot)$ is differentiable (β-Lipschitz continuous)
- $\Gamma_0(\mathbb{R}^L)$: Set of proper semi-lower-continuous convex functions

【Example】

- $f(\mathbf{s}) = \frac{1}{2}\|\mathbf{v} - \mathbf{D}\mathbf{s}\|_2^2$
- $g(\mathbf{s}) = \lambda\|\mathbf{s}\|_1$

関数プロット (Function plot)

```
% Function setting
fg = @(s0,s1) 0.5*(v-(D(1)*s0+D(2)*s1)).^2 + lambda*(abs(s0)+abs(s1));
% Surfc plot of cost functio f+g
figure(2)
J = fg(S0,S1);
hf = surfc(s0,s1,J);
hf(1).FaceAlpha = 0.25;
hf(1).EdgeAlpha = 0.25;
hf(1).EdgeColor = 'interp';
hf(2).LineWidth = 1;
set(gca,'YDir','reverse')
ylabel('s_0')
xlabel('s_1')
zlabel('f(s)+g(s)')
hold on
```



ネステロフの加速法
(Nesterov's acceleration)

1. Initialization: $\mathbf{y}^{(1)} = \mathbf{x}^{(0)}$, $a^{(1)} = 1$, $t \leftarrow 0$
2. Proximal gradient descent: $\mathbf{x}^{(t+1)} \leftarrow \mathrm{prox}_{\gamma g}\left(\mathbf{y}^{(t)} - \gamma \nabla_{\mathbf{x}} f(\mathbf{y}^{(t)})\right)$
3. $a^{(t+1)} \leftarrow \dfrac{1+\sqrt{1+(a^{(t)})^2}}{2}$
4. $\mathbf{y}^{(t)} \leftarrow \mathbf{x}^{(t)} + \dfrac{a^{(t)}-1}{a^{(t+1)}}(\mathbf{x}^{(t)} - \mathbf{x}^{(t-1)})$
5. If a stopping critera is satisfied then finish, otherwise $t \to t+1$ and go to Step 2.

【Example】

- $\nabla_{\mathbf{s}} f(\mathbf{s}) = \mathbf{D}^T(\mathbf{D}\mathbf{s} - \mathbf{v})$
- $\mathrm{prox}_{\gamma\lambda\|\cdot\|_1}(\mathbf{s}) = \mathcal{T}_{\gamma\lambda}(\mathbf{s}) = \mathrm{sign}(\mathbf{s}) \odot \max(\mathrm{abs}(\mathbf{s}) - \gamma\lambda\mathbf{1}, \mathbf{0})$

ソフト閾値処理 (Soft-thresholding)

```
softthresh = @(x,t) sign(x).*max(abs(x)-t,0);
```

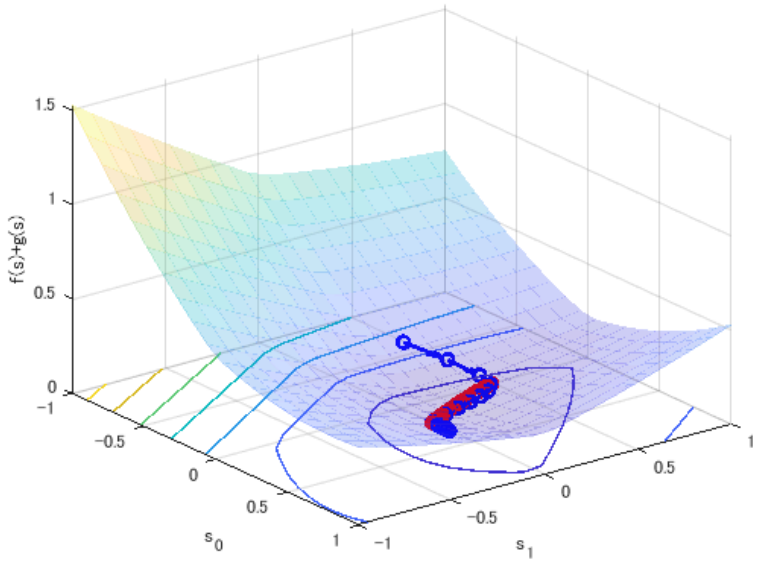
初期化 (Initialization)

```
sp0 = 2*rand(2,1)-1; % in [-1,1]^2
```

近接勾配降下 (Proximal gradient descent)

```
beta = D*D';
assert(gamma < 2/beta,'Step size condition is violated.')
```

```
for isaxel = [ false, true ]
    ac = 1;
    sp = sp0;
    z = sp0;
    for idx=1:niters
        % Preious state
        s(1,1) = sp(1); % s0
        s(2,1) = sp(2); % s1
        % Proximal gradient descent
        sc = softthresh(z-gamma*D'*(D*z-v),gamma*lambda);
        %
        if isaxel
            an = (1+sqrt(1+4*ac^2))/2;
        else
            an = 1;
        end
        %
        z = sc + (ac-1)/an*(sc-sp);
        % Current state
        s(1,2) = sc(1); % s0
        s(2,2) = sc(2); % s1
        % Quiver plot
        xp = s(2,1);
        yp = s(1,1);
        xn = s(2,2);
        yn = s(1,2);
        hp = quiver(xp,yp,xn-xp,yn-yp);
        hp.Marker = 'o';
        hp.ShowArrowHead = 'on';
        hp.MaxHeadSize = 120;
        hp.MarkerSize = 6;
        if isaxel
            hp.MarkerEdgeColor = 'b';
            hp.Color = 'b';
        else
            hp.MarkerEdgeColor = 'r';
            hp.Color = 'r';
        end
        hp.LineWidth = 2;
        % Update
        sp = sc;
        ac = an;
    end
end
hold off
```



パラメータ設定
(Parameter settings)

- sgm: ノイズ標準偏差 σ_w (Standard deviation of noise)
- nlevels: ウェーブレット段数 (Wavelet levels)

```
% Parameter settings
isaprxleft = true;
lambda = 10^-1% -0.1
```

```
lambda = 0.1000
```

```
gamma = 10^0.1% -0.8
```

```
gamma = 1.2589
```

```
sgmuint8 = 10;
sgm = sgmuint8/255;
nlevels = 3;
niters = 40;
```

画像の読込
(Read image)

```
u = rgb2gray(im2double(imread('./data/lena.png')));
```

観測画像
(Observation image)

- $\mathbf{v} = \mathbf{P}\mathbf{u} + \mathbf{w}$
- $\mathbf{u} = \mathbf{D}\mathbf{s}$
- $\mathbf{w} \sim \mathrm{Norm}\left(\mathbf{w} | \boldsymbol{\mu}_w = \mathbf{0}, \sigma_w^2 \mathbf{I}\right)$

```
% Definition of measurment process
psf = fspecial('motion',21,11);
measureproc = @(x) imfilter(x,psf,'conv','circular');
% Adjoint process of the measurment process
measureadjp = @(x) imfilter(x,psf,'corr','circular');
% Simulation of AWGN
v = imnoise(measureproc(u),'gaussian',0,sgm^2);
```

非間引きハール**DWT**

(Undecimated Haar DWT)

```
import msip.udhaarwtdec2
import msip.udhaarwtrec2
```

完全再構成の確認 **(Checki the perfect reconstruction)**

非間引きハールDWTはパーセバルタイト性 (The undecimated DWT satisfies the Parseval tight property,)

$$\mathbf{D}\mathbf{D}^T = \mathbf{I}$$

を満たすため， $\mathbf{D}$ の転置システムは完全再構成分析システムとなり得る．(and thus Its transposition system can be a PR analysis system.)

```
[coefs,scales] = udhaarwtdec2(v,nlevels);
r = udhaarwtrec2(coefs,scales);
assert(norm(v-r,"fro")^2/numel(v)<1e-18,'Perfect reconstruction is violated.')
```

合成辞書と転置辞書の定義 (Definition of synthesis dictionary and its adjoint)

```
% Definiton of dictionay and its adjoint
adjdic = @(x) udhaarwtdec2(x,nlevels); % D
syndic = @(x) udhaarwtrec2(x,scales);  % D.'
```

高速**ISTA**

(Fast ISTA)

問題設定 **(Problem setting)**

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}} \frac{1}{2} \|\mathbf{v} - \mathbf{P}\mathbf{D}\mathbf{s}\|_2^2 + \lambda \|\mathbf{s}\|_1$$

アルゴリズム **(Algorithm)**

1. Initialization: $\mathbf{y}^{(1)} = \mathbf{s}^{(0)}$, $a(1) = 1$, $t \leftarrow 0$
2. $\mathbf{s}^{(t+1)} \leftarrow \mathrm{prox}_{\gamma g}\left(\mathbf{y}^{(t)} - \gamma \nabla_{\mathbf{s}} f\left(\mathbf{y}^{(t)}\right)\right)$

3. $a^{(t+1)} \leftarrow \frac{1+\sqrt{1+4(a^{(t)})^2}}{2}$
4. $\mathbf{y}^{(t)} \leftarrow \mathbf{s}^{(t)} + \frac{a^{(t)}-1}{a^{(t+1)}}(\mathbf{s}^{(t)} - \mathbf{s}^{(t-1)})$
5. If a stopping critera is satisfied then finish, otherwise $t \rightarrow t+1$ and go to Step 2.

ただし，(where)

- $\nabla_{\mathbf{s}} f(\mathbf{s}) = \mathbf{D}^T\mathbf{P}^T(\mathbf{PDs} - \mathbf{v})$
- $\mathrm{prox}_{\gamma\lambda\|\cdot\|_1}(\mathbf{s}) = \mathcal{T}_{\gamma\lambda}(\mathbf{s}) = \mathrm{sign}(\mathbf{s}) \odot \max(\mathrm{abs}(\mathbf{s}) - \gamma\lambda\mathbf{1}, \mathbf{0})$

ソフト閾値処理 (Soft-thresholding)

```
softthresh = @(x,t) sign(x).*max(abs(x)-t,0);
```

初期化 (Initialization)

```
[coefs,scales] = udhaarwtdec2(v,nlevels);
sp = coefs;
```

近接勾配降下 (Proximal gradient descent)

- $\gamma < 2/\beta$: Step size
- β: Lipschitz constant of ∇f, where $\beta = (\sigma_{\max}(\mathbf{PD}))^2$

```
beta = max(abs(fftn(psf,2.^nextpow2(size(v)))),[],'all');
assert(gamma < 2/beta,'Step size condition is violated.')
if isaprxleft
    mask = ones(size(coefs));
    mask(1:prod(scales(1,:))) = 0;
    lambda = lambda * mask;
end
yc = sp;
ac = 1;
for idx=1:niters
    % Proximal gradient descent
    sg = adjdic(measureadjp(measureproc(syndic(yc))-v));
    sc = softthresh(yc-gamma*sg,gamma*lambda);
    an = (1+sqrt(1+4*ac^2))/2;
    yc = sc + ((ac-1)/an)*(sc-sp);
    % Update
    sp = sc;
    ac = an;
end
```

復元画像
(Restored image)

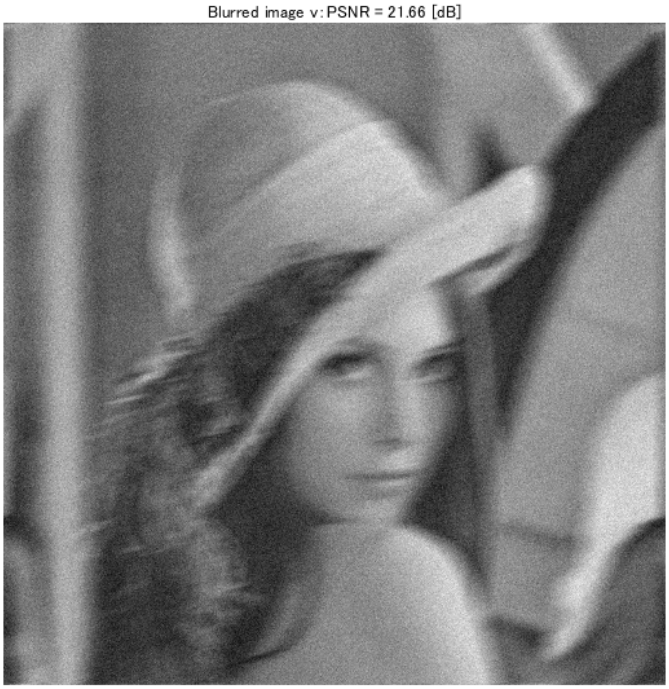
```
r = syndic(sc);
```

画像表示

(Image show)

```
figure(1)
imshow(u);
title('Original image u')
```


```
figure(2)
imshow(v)
title(sprintf('Blurred image v : PSNR = %5.2f [dB]',psnr(u,v)))
```




```
figure(3)
imshow(r)
title(sprintf('Restored image r w/ FISTA：PSNR = %5.2f [dB]',psnr(u,r)))
```