Sample 5-6

周波数解析

多変量循環畳み込み

画像処理特論

村松 正吾

動作確認: MATLAB R2023a

Fourier analysis

Multivariate circular convolution

Advanced Topics in Image Processing

Shogo MURAMATSU

Verified: MATLAB R2023a

準備

(Preparation)

```
close all
```

サンプル画像の準備

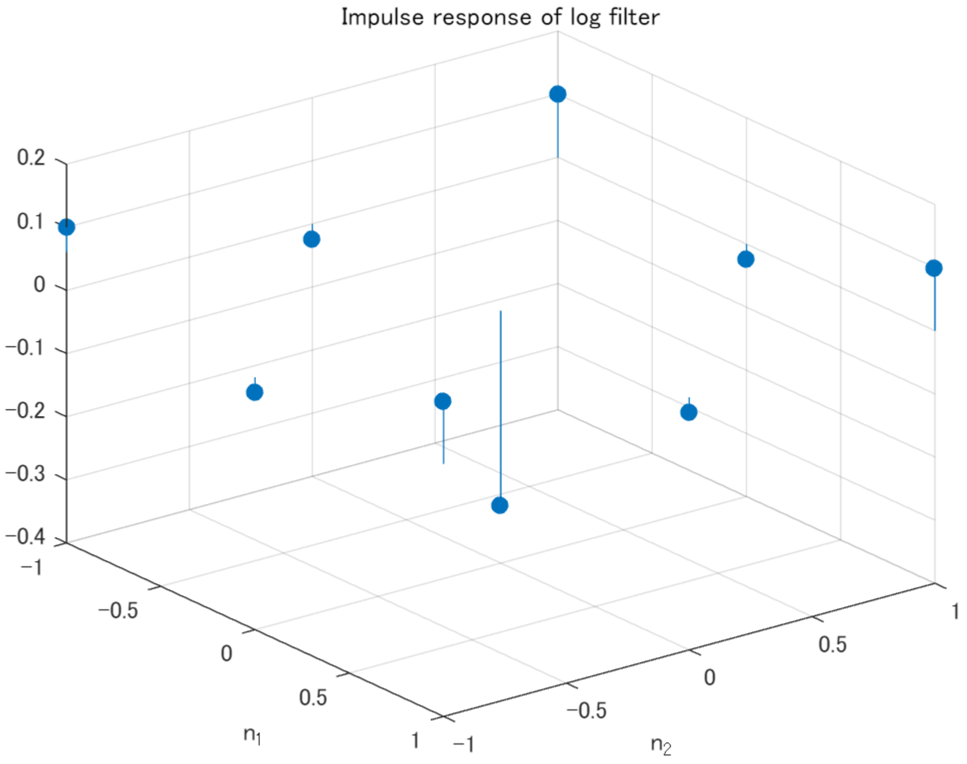
(Preparation of sample image)

```
% Reading original image
u = im2double(imread('cameraman.tif'));
figure(1)
imshow(u)
title('Original')
```

線形シフト不変システムのインパルス応答

(Impulse response of a linear shift-invariant system

```
% Impulse response h[n]
hsize1 = 3;
hsize2 = 3;
sigma = 1;
ftype = "log";
h = rot90(fspecial(ftype,[hsize1 hsize2],sigma),2);
figure(2)
[n1,n2] = meshgrid(-floor((hsize2-1)/2):ceil((hsize2-1)/2),-floor((hsize1-1)/
2):ceil((hsize1-1)/2));
stem3(n1,n2,h,'filled')
xlabel('n_2')
ylabel('n_1')
axis ij
title(['Impulse response of ' char(ftype) ' filter'])
```



周期行列 の循環畳み込みの出力応答

(Output response of circular convolution with period)

循環畳み込み演算 (Circular convolution)

```
% Setting the period N
nPeriod1 =258;
```

```
nPeriod2 =258;
nPeriod = [nPeriod1 nPeriod2];
nZeroPadding = [nPeriod1 nPeriod2] - size(u);

% Zero padding
uzpd = padarray(u,nZeroPadding,0,'post');
figure(3)
imshow(uzpd)
```

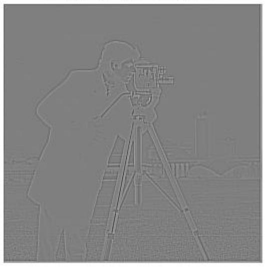
```


% Output v[n]
v = imfilter(uzpd,h,'conv','circ');
```

畳み込み演算との比較

(Comparison with convolution)

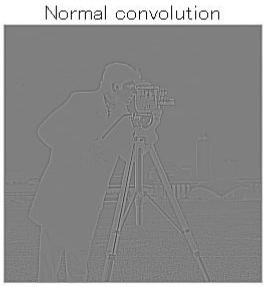
```
% Normal convolution
w = imfilter(u,h,'conv','full');

% v[n]
figure(4)
imshow(v+(min(v(:))<0)/2)
title('Circular convolution')
```

Circular convolution



```
% w[n]
figure(5)
```

```
imshow(w+(min(w(:))<0)/2)
title('Normal convolution')
```



通常の畳み込みと循環畳み込みが一致する条件

(The condition that normal convolution and circular convolution match)

ただし、(where)

- : 出力のサポート領域 (Output support region)
- : インパルス応答のサポート領域 (Support region of impulse response)
- : 入力のサポート領域 (Input support region)

以下では周期行列 を対角行列 (In the following, the periodic matrix is set to a diagonal matrix)

に設定する．すなわち，(That is,)

ただし， を仮定する．(and is assumed.)

【Example】もし，(If)

ならば，(then,)

よって，(Therefore, from)

より，

ならば，通常と畳み込みと循環畳み込みの結果が一致する．(then, the results of normal, convolution and circular convolution are consistent.)

```
% Adjusting the sizes for evaluation
dsz = size(v)-size(w);
if dsz(1) > 0
    vc = v;
    wc = padarray(w,[dsz(1) 0],0,'post');
else
    wc = w;
    vc = padarray(v,[-dsz(1) 0],0,'post');
end
if dsz(2) > 0
    wc = padarray(wc,[0 dsz(2)],0,'post');
else
    vc = padarray(vc,[0 -dsz(2)],0,'post');
end
% Compensate the circular shift
wc = circshift(wc,-ceil((size(h)-1)/2));

% Sizes and MSE
mymse = @(x,y) mean((double(x)-double(y)).^2,'all');
fprintf('Period:         N1  = %d, N2  = %d',nPeriod1,nPeriod2);
```

```
Period:         N1  = 258, N2  = 258
```

```
fprintf('Size of image:  Lu1 = %d, Lu2 = %d',size(u,1),size(u,2));
```

```
Size of image:  Lu1 = 256, Lu2 = 256
```

```
fprintf('Size of filter: Lh1 = %d, Lh2 = %d',size(h,1),size(h,2));
```

```
Size of filter: Lh1 = 3, Lh2 = 3
```

```
fprintf('MSE: %f', mymse(vc,wc))
```

```
MSE: 0.000000
```

入力信号 の DFT

(DFT of input signal)

```
% DFT of u[n]
U = fftn(u,nPeriod);
```

フィルタ の DFT

(DFT of impulse response)

```
% DFT of h[n]
H = fftn(h,nPeriod);
```

出力信号 の DFT

(DFT of output signal)

```
% Frequency response of v[n]
V = fftn(v,nPeriod);
```

DFT 積

(DFT product)

循環畳み込みとの比較 (Comparison with circular convolution)

```
% IDFT of DFT product
y = ifftn(H.*U);
% Compensate the circular shift
y = circshift(y,-ceil((size(h)-1)/2));
% MSE with the cconv result 'v'
fprintf('MSE: %f', mymse(v,y))
```

```
MSE: 0.000000
```

循環畳み込みのスペクトルノルム

(Spectral norm of the circular convolution)

ただし，(where)

- : の最大特異値．（Maximum singular value of)

```
% Definition of map T as a circular convolution with h[n]
mapT = @(x) imfilter(x,h,'conv','circ');
```

2 変量循環畳み込みの行列表現

(Matrix representation of the bivariate circular convolution)

```
% Redefining the period
N1 =8;
N2 =8;

% Find the matrix representation of the circular convolution
```

```
N  = N1*N2;
T = zeros(N);
for idx = 1:N
    % Generating a standard basis vector
    e = zeros(N1,N2);
    e(idx) = 1;
    % Response to the standard basis vector
    T(:,idx) = reshape(mapT(e),[N 1]);
end
% Matrix representation of the circular convolution
T
```

```
T = 64×64
   -0.3079   -0.0234         0         0         0         0         0   -0.0234 ···
   -0.0234   -0.3079   -0.0234         0         0         0         0         0
         0   -0.0234   -0.3079   -0.0234         0         0         0         0
         0         0   -0.0234   -0.3079   -0.0234         0         0         0
         0         0         0   -0.0234   -0.3079   -0.0234         0         0
         0         0         0         0   -0.0234   -0.3079   -0.0234         0
         0         0         0         0         0   -0.0234   -0.3079   -0.0234
   -0.0234         0         0         0         0         0   -0.0234   -0.3079
   -0.0234    0.1004         0         0         0         0         0    0.1004
    0.1004   -0.0234    0.1004         0         0         0         0         0
      ⋮
```

スペクトルノルム

(Spectral norm)

```
% Function NORM evaluates the operator norm for a matrix
opnorm = norm(T,2)
```

```
opnorm = 0.7096
```

最大特異値

(Maximum singular value)

```
sigma1 = max(svd(T))
```

```
sigma1 = 0.7096
```

最大振幅応答

(Maximum magnitude response)

```
H = fftn(h,[N1 N2]);
maxmgn = max(abs(H(:)))
```

```
maxmgn = 0.7096
```

関数 NORM に関する注意

(Notes on the Function NORM)

行列に関するノルムを評価する際には引数の渡し方、オプションの指定に注意すること．(When evaluating the entorywise norm of a matrix, pay attention to the way of passing the arguments and options.)

```
% Froubenius norm
norm(T,'fro')
```

```
ans = 2.9649
```

```

% Entrywise 2-norm, which is identical to the Frobenius norm
norm(T(:),2)
```

```
ans = 2.9649
```

```

% Operator 1-norm
norm(T,1)
```

```
ans = 0.8033
```

```

% Entrywise 1-norm
norm(T(:),1)
```

```
ans = 51.4119
```