Inverse Problems Exercises: 2024s s04 (non-sc)

https://www.umm.uni-heidelberg.de/miism/

Notes

- Please **DO NOT** change the name of the `.ipynb` file.
- Please **DO NOT** import extra packages to solve the tasks.
- Please put the `.ipynb` file directly into the `.zip` archive without any intermediate folder.

Please provide your personal information

- full name (Name):

YOUR ANSWER HERE

D07: Singular value decomposition

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt

from numpy import linalg as LA
```

In [ ]:
```
file_gaussian = 'file_gaussian.npz'
with np.load(file_gaussian) as data:
    f_true = data['f_true']
    A_psf = data['A_psf']
    list_gn = data['list_gn']
```

Imaging model

The imaging model can be represented by

$$g = h \otimes f_{\text{true}} = Af_{\text{true}} = \mathcal{F}^{-1}\{\mathcal{F}\{h\}\mathcal{F}\{f_{\text{true}}\}\},$$
$$g' = g + \epsilon.$$

- f_{true} is the input signal
- h is the point spread function (kernel)
- $\otimes$ is the convolution operator
- A is the Toeplitz matrix of h
- $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the Fourier transform operator and inverse Fourier transform operator
- ϵ is the additive Gaussian noise
- g is the filtered signal
- g' is the noisy signal

Frobenius norm

Implement the Frobenius norm by definition

$$\|A\|_{\text{F}} = \sqrt{\sum_{i}^{m}\sum_{j}^{n}|a_{ij}|^2},$$

where A is an $m \times n$ matrix. When $m = 1$ or $n = 1$, A is a vector.

- Given the matrix A
- Calculate the Frobenius norm (`numpy.linalg.norm()` should NOT be used.)
- Implement the function `frobenius_norm()`

In [ ]:
```python
def frobenius_norm(mat_A):
    """ Compute the Frobenius norm of the matrix:

    :param mat_A: Input matrix or vector.
    :returns: Frobenius norm.
    """
# YOUR CODE HERE
raise NotImplementedError()
```

In [ ]:
```python
# This cell contains hidden tests.
```

Condition number of A

Calculate the condition number of matrix A

$$\mathrm{cond}_{\mathrm{F}}(A) = \|A^{-1}\|_{\mathrm{F}} \|A\|_{\mathrm{F}}$$

- Apply the calculation to `A_psf`
- Save $\|A\|_{\mathrm{F}}$ in the variable `norm_A_psf` (using `frobenius_norm()`)
- Save $\|A^{-1}\|_{\mathrm{F}}$ in the variable `norm_A_inv` (using `frobenius_norm()`)
- Save $\mathrm{cond}_{\mathrm{F}}(A)$ in the variable `cond_A_psf` (`numpy.linalg.cond()` should not be used.)

In [ ]:
```
# YOUR CODE HERE
raise NotImplementedError()
```

In [ ]:
```
# This cell contains tests.

print(cond_A_psf)
```

Question: Condition number of A

Is the inversion of the system with `A_psf` stable?

YOUR ANSWER HERE

Singular value decomposition

Calculate the singular value decomposition (SVD) of real matrix A

$$A = USV^T,$$

where $UU^T = VV^T = I$, and S is a rectangular diagonal matrix with non-negative real numbers on the diagonal. The entries on the diagonal $\mathrm{diag}(S)_i$ are the singular values.

- Apply the calculation to `A_psf` (using `numpy.linalg.svd()`)
- Plot A, U, S and V^T in order in the subplots of `axs`
- Plot the matrices A, U, and V^T as grayscale images in the corresponding subplots
- Show the colorbar of the above three subplots
- Plot the singular values $\mathrm{diag}(S)_i$ as a line in the third subplot
- Add proper titles to the subplots of `axs`

In [ ]:
```
fig, axs = plt.subplots(1, 4, figsize=(15, 5))
fig.suptitle('Singular value decomposition')

# YOUR CODE HERE
raise NotImplementedError()
```

In [ ]:
```
# This cell contains hidden tests.
```

Condition number of S

Calculate the condition number of matrix S

- Apply the calculation to the S from SVD
- Save $||S||_{\mathrm{F}}$ in the variable `norm_S_psf` (using `frobenius_norm()`)
- Save $||S^{-1}||_{\mathrm{F}}$ in the variable `norm_S_inv` (using `frobenius_norm()`)
- Save $\mathrm{cond}_{\mathrm{F}}(S)$ in the variable `cond_S_psf` (`numpy.linalg.cond()` should not be used.)

In [ ]:
```
# YOUR CODE HERE
raise NotImplementedError()
```

In [ ]:
```
# This cell contains tests.

print(cond_S_psf)
```

Question: Condition number of S

Is the equation $\mathrm{cond}_{\mathrm{F}}(A) = \mathrm{cond}_{\mathrm{F}}(S)$ valid? Why?

YOUR ANSWER HERE

Truncated singular values

S_t contains the truncated singular values as

$$\mathrm{diag}(S_t)_i = \begin{cases} \mathrm{diag}(S)_i & \mathrm{diag}(S)_i \geq \mathrm{TH} \\ 1 & \text{otherwise} \end{cases},$$

i.e. to set the singular values in S less than the threshold TH to 1.

- Set TH with $2\% \cdot \max(\mathrm{diag}(S)_i)$, $10\% \cdot \max(\mathrm{diag}(S)_i)$, $50\% \cdot \max(\mathrm{diag}(S)_i)$, respectively
- Generate S_t
- Save $A_t = US_tV^T$ in the variable `list_A_tsvd` (as `list` of `numpy.array`)
- Save $\mathrm{cond}_{\mathrm{F}}(A_t)$ in the variable `list_cond_A_tsvd` (`numpy.linalg.cond()` should not be used.)

In [ ]:
```
# YOUR CODE HERE
raise NotImplementedError()
```

In [ ]:
```
# This cell contains tests.

print(list_cond_A_tsvd)
```

Reconstruction

Reconstruct the signal by

$$\tilde{f} = A_t^{-1} g'$$

- Apply this operation to the noisy signals in `list_gn`
- Return the outputs with different A_t in `list_A_tsvd`
- Save the outputs in the variable `list_f_tsvd` (as `list` of `numpy.array`)

Display the result

- Plot the outputs in `list_f_tsvd` in the same order of the noisy signals in the subplots of `axs`
- Show the cases of the same noisy signal in the same subplot column
- Show the cases with the same A_t in the same subplot row
- Plot the corresponding noisy signal in each subplot (after the filter output)
- Plot the input signal `f_true` in each subplot (after the noisy signal)
- Show the legend in each subplot
- Show the case information in the titles to the subplots

In [ ]:

```
fig, axs = plt.subplots(3, 3, figsize=(15, 15))
fig.suptitle('Reconstruction')

# YOUR CODE HERE
raise NotImplementedError()
```

In [ ]:

```
# This cell contains hidden tests.
```

Question: Truncated singular values

Describe the visual effect on the reconstruction result considering the influence of TH.

YOUR ANSWER HERE