EE5147 Modern Spectral Analysis
Final Project: Spectrum Estimation with Compressive Sensing

Notice

- $\boxed{\textbf{Due at 9:00pm, June 17, 2022 (Friday)}}$ $= T_d$ for the electronic copy of your final report.
- Please submit **your report (.pdf file)**, **all your MATLAB files (.m files)**, and **your results (.mat file)** to NTU COOL (`https://cool.ntu.edu.tw/courses/11382`)
- Please **type your final report**. This report can be typed either in English or in Chinese.
- No extensions, unless granted by the instructor one day before T_d.

Introduction

This course introduced several spectrum estimators, such as the spectrogram, the MVDR spectrum, the MUSIC algorithm, and the ESPRIT algorithm. In the second part of this course, we introduced compressive sensing, which can be regarded as solving a underdetermined systems with sparse solutions. In this final project, we will apply compressive sensing to spectrum estimation.

Data Model and Problem Formulation

We consider random process $x(n)$ with the following model

$$x(n) = \sum_{p=1}^{P} \sigma_p e^{j2\pi f_p n} s_p(n) + w(n). \tag{1}$$

where P is the number of complex exponentials. The pth signal term in (1) has magnitude σ_p, frequency f_p, and signal $s_p(n)$. The noise term is denoted by $w(n)$. The time index n belongs to the set $\{0, 1, \ldots, N-1\}$. We assume some prior knowledge about (1):

- The number of complex exponentials is deterministic and known to be $P = 5$.
- The magnitude $\sigma_p > 0$ is deterministic but unknown.
- The frequency $f_p \in [-1/2, 1/2)$ is deterministic but unknown.
- The signal $s_p(n)$ is deterministic and known to be $s_p(n) = e^{j\pi/6}$ for all p and n.
- The noise term is a circularly-symmetric complex Gaussian random variable with properties $\mathbb{E}[w(n)] = 0$ (zero mean) and $\mathbb{E}[w(n)w^*(n-k)] = \sigma_w^2\delta(k)$ (white).
- The noise variance $\sigma_w^2 > 0$ is deterministic but unknown.

Design Objective

We denote $\widetilde{x}(n)$ for $n \in \{0, 1, \ldots, N\}$ as a *realization* of $x(n)$ in (1). Based on $\widetilde{x}(n)$, design the following estimators:

1. An estimator for the magnitudes $\widehat{\sigma}_1, \widehat{\sigma}_2, \ldots, \widehat{\sigma}_P$.
2. An estimator for the frequencies $\widehat{f}_1, \widehat{f}_2, \ldots, \widehat{f}_P$.
3. An estimator for the noise standard deviation $\widehat{\sigma}_w$.

Design Requirements

1. **Your estimators should be based on the algorithms for compressive sensing** in `09_Basic_Algorithm_in_CS.pdf`.
2. The estimated magnitudes are positive numbers ($\widehat{\sigma}_1, \widehat{\sigma}_2, \ldots, \widehat{\sigma}_P > 0$).
3. The estimated frequencies are sorted in the ascending order ($-\frac{1}{2} \leq \widehat{f}_1 \leq \widehat{f}_2 \leq \cdots \leq \widehat{f}_P < \frac{1}{2}$).
4. The estimated noise standard deviation is positive ($\widehat{\sigma}_w > 0$).

Grading

The grading for the final project is as follows:

- **Final report (65%):**
  - **Motivation:** Review some existing methods applicable to this problem. Among these methods, explain the reason why your estimator is selected.
  - **Descriptions of the estimators:** Describe the details of your estimators using math equations.
  - **Monte-Carlo experiments**: Please design your own Monte-Carlo experiments to answer the following questions:
    * Why is the proposed estimator applicable to the data model in (1)?
    * If the proposed estimators have some hyperparameters, how do you select their numerical values?

    Hints: Recall the Monte-Carlo experiments in HW1, HW2, and HW3. You need to first assume the true parameters, generate your own data according to (1), apply the estimators, and finally evaluate the performance metrics.
- **Executable MATLAB codes (10%)**
- **Performance evaluation (25%)**

Performance Evaluation Based on Testing Data (25%)

The testing data is provided in `MSA_Final.mat`, where you can find one realization of (1). This realization is represented in the following vector.

$$\texttt{x\_tilde} \triangleq \begin{bmatrix} \widetilde{x}(0) & \widetilde{x}(1) & \ldots & \widetilde{x}(N-1) \end{bmatrix}. \tag{2}$$

Please apply your estimators to this realization.

We will assess your estimation performance according to the output of your estimators. **Please submit your data in a mat file**. This file contains three vectors, defined as

$$\texttt{sigma\_hat} \triangleq \begin{bmatrix} \widehat{\sigma}_1 & \widehat{\sigma}_2 & \ldots & \widehat{\sigma}_P \end{bmatrix}, \tag{3}$$

$$\texttt{f\_hat} \triangleq \begin{bmatrix} \widehat{f}_1 & \widehat{f}_2 & \ldots & \widehat{f}_P \end{bmatrix}, \tag{4}$$

$$\texttt{sigma\_w\_hat} \triangleq \widehat{\sigma}_w, \tag{5}$$

where the estimated magnitudes and estimated frequencies are $\widehat{\sigma}_p$, and $\widehat{f}_p$, respectively. The estimated noise standard deviation is $\widehat{\sigma}_w$.

We will compare your results with the *true* parameters Let the vectors associated with the true parameters be

$$\texttt{sigma\_true} \triangleq \begin{bmatrix} \sigma_1 & \sigma_2 & \ldots & \sigma_P \end{bmatrix}, \tag{6}$$

$$\texttt{f\_true} \triangleq \begin{bmatrix} f_1 & f_2 & \ldots & f_P \end{bmatrix}, \tag{7}$$

$$\texttt{sigma\_w\_true} \triangleq \sigma_w, \tag{8}$$

But **these true answers are unavailable to you**. We will evaluate your grade in this part by the MATLAB script

```
% sigma_hat: 10%
max([ round( 10 * ( 1 - norm(sigma_hat(:) - sigma_true(:)) / norm(sigma_true) ) ), 0])
% f_hat: 10%
max([ round( 10 * ( 1 - norm(f_hat(:) - f_true(:)) / norm(f_true) ) ), 0])
% sigma_w_hat: 5%
max([ round( 5 * ( 1 - norm(sigma_w_hat(:) - sigma_w_true(:)) / norm(sigma_w_true) ) ), 0])
```

In other words, if your results are closer to the true parameters, then you get more grades.

Last updated June 2, 2022.