MSA Final Project - Spectrum Estimation with Compressive Sensing

Motivation

The data model is $x(n) = \sum_{p=1}^{P} \sigma_p e^{j2\pi f_p n} s_p(n) + w(n)$. We know that $s_p(n) = e^{j\pi/6}$; therefore, $\tilde{x}(n) = \sum_{p=1}^{P} \sigma_p e^{j2\pi f_p n} + \tilde{w}(n)$ where $\tilde{x}(n) = e^{-j\pi/6} x(n)$ and $\tilde{w}(n) = e^{-j\pi/6} w(n)$. Notice that $\tilde{w}(n)$ is still white gaussian with the same statistical characteristic.

Rewrite the above equation in matrix form, we have:

$$\begin{bmatrix} \tilde{x}(1) \\ \tilde{x}(2) \\ \vdots \\ \tilde{x}(N) \end{bmatrix} = \begin{bmatrix} \mathbf{v}(f_1) & \mathbf{v}(f_2) & \dots & \mathbf{v}(f_G) \end{bmatrix} \begin{bmatrix} \sigma_{sparse}(1) \\ \sigma_{sparse}(2) \\ \vdots \\ \sigma_{sparse}(G) \end{bmatrix} + \begin{bmatrix} \tilde{w}(1) \\ \tilde{w}(2) \\ \vdots \\ \tilde{w}(N) \end{bmatrix}$$

where N is number of samples, G is the frequency grid. Notice that $\boldsymbol{\sigma}_{sparse}$ is P-sparse; therefore, we can apply the compressive sensing methods for the estimation.

We define $\mathbf{A} = \begin{bmatrix} \mathbf{v}(f_1) & \mathbf{v}(f_2) & \dots & \mathbf{v}(f_G) \end{bmatrix}$.

Descriptions of the estimators

This project implements four compressive sensing algorithms: Orthogonal Matching Pursuit (OMP), Compressive Sampling Matching Pursuit (CoSaMP), Iterative Hard Threshold (IHT) and Hard Thresholding Pursuit (HTP).

OMP

(1) Project Residual on Basis.

> $\mathbf{r_p} = \mathbf{A}^H(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\sigma}_{sparse})$

(2) Find largest Projection Basis

> $i = \arg\max(\mathbf{r_p})$

(3) Add it to the support

> $supp(\boldsymbol{\sigma}_{sparse}) \leftarrow supp(\boldsymbol{\sigma}_{sparse}) \cup \{i\}$

(4) Find the LS solution corresponding to the given support. Notice that we take the real part since $\boldsymbol{\sigma}$ is a real-valued vector ($\mathbf{M}^{\#}$ denotes the pseudo inverse of $\mathbf{M}$)

> $\boldsymbol{\sigma}_{sparse\,supp(\boldsymbol{\sigma}_{sparse})} = \Re(\mathbf{A}^{\#}_{:,supp(\boldsymbol{\sigma}_{sparse})}\tilde{\mathbf{x}})$

(5) Repeat (1)-(4) for s times.

CoSaMP

(1) Project Residual on Basis.

> $\mathbf{r_p} = \mathbf{A}^H(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\sigma}_{sparse})$

(2) Find $2s$ largest Projection Basis

> $\mathcal{I} = \arg\max(\mathbf{r_p}, 2s)$

(3) Add it to the support

> $supp(\boldsymbol{\sigma}_{sparse}) \leftarrow supp(\boldsymbol{\sigma}_{sparse}) \cup \mathcal{I}$

(4) Find the LS solution corresponding to the given support.

> $\boldsymbol{\sigma}_{sparse\,supp(\boldsymbol{\sigma}_{sparse})} = \Re(\mathbf{A}^{\#}_{:,supp(\boldsymbol{\sigma}_{sparse})}\tilde{\mathbf{x}})$

(5) Find s largest magnitude on $\boldsymbol{\sigma}_{sparse}$ leave them and set the others as zero

> $\mathcal{J} = \arg\max(\mathbf{r_p}, s)$
> $\sigma_{sparse\,\{1,\dots,G\}/\mathcal{J}} \leftarrow 0$

(6) Repeat (1)-(5) for s times.

IHT

(1) Update $\boldsymbol{\sigma}_{sparse}$ by projection on basis.

> $\boldsymbol{\sigma}^{(next)}_{sparse} = \boldsymbol{\sigma}_{sparse} + \frac{1}{G}\mathbf{A}^H(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\sigma}_{sparse})$

(2) Find s largest value in $\boldsymbol{\sigma}_{sparse}$ leave them and set the others as zero

> $\mathcal{I} = \arg\max(\boldsymbol{\sigma}^{(next)}_{sparse}, s)$
> $\sigma^{(next)}_{sparse\,\{1,\dots,G\}/\mathcal{I}} \leftarrow 0$

> $\boldsymbol{\sigma}_{sparse} \leftarrow \boldsymbol{\sigma}_{sparse}^{(next)}$

(3) Repeat (1)-(2) for s times.

HTP

(1) Update $\boldsymbol{\sigma}_{sparse}$ by projection on basis.

> $\boldsymbol{\sigma}_{sparse}^{(next)} = \boldsymbol{\sigma}_{sparse} + \frac{1}{G}\mathbf{A}^H(\tilde{\mathbf{x}} - \mathbf{A}\boldsymbol{\sigma}_{sparse})$

(2) Find s largest value in $\boldsymbol{\sigma}_{sparse}$

> $\mathcal{I} = \arg\max(\boldsymbol{\sigma}_{sparse}^{(next)}, s)$

(3) Find LS Solution baesd on the support leave them and set the others as zero

> $\boldsymbol{\sigma}_{sparse_{\mathcal{I}}} = \Re(\mathbf{A}_{:,\mathcal{I}}^{\#}\tilde{\mathbf{x}})$
> $\sigma_{sparse}{}_{\{1,\ldots,G\}/\mathcal{I}}^{(next)} \leftarrow 0$

(4) Repeat (1)-(3) for s times.

Monte-Carlo experiments

The number of grids for each algorithm is select as follows:

Algorithm	Grid Size
OMP	1000
CoSaMP	200
IHT	200
HTP	200

We find that for CoSaMP, IHT and HTP, when the Grid Size is set larger than 200, it has a large probability to suffer severe degradation. This might results from parllel finding s sparse solution in one single iteration.

The simulation data is generated by the following statistical model:

Var.	Statistical Model

Var.	Statistical Model
$\mathbf{f}$	uniform [-0.5,0.5] & independent
$\boldsymbol{\sigma}$	uniform [0.5,1.5] & independent
σ_w	uniform [0.1,1.5]

The following table shows the average score for each algorithm in the monte-carlo runs (run for 1000 simulation):

	OMP	CoSaMP	IHT	HTP
Sigma Score (0~10)	8.4020	5.9340	7.2490	5.8030
F Score (0~10)	9.6030	7.9890	7.6740	6.4440
Sigma_w Score (0~5)	4.7240	2.9440	3.5640	3.2790

It is clear that OMP domainates other three algorithm in all three scores; therefore, we choose to estimate the data by the OMP algorithm.

How to run?

(1) Run "top.m" for generating the estimating results
(2) Run "test.m" to run monte-carlo simulations on all algorithms

File Description

- top.m: run OMP and generate results
- test.m: monte-carlo runs for all algorithm
- OMP.m: Orthogonal Matching Pursuit Algorithm
- CoSaMP.m: Compressive Sampling Matching Pursuit
- IHT.m: Iterative Hard Threshold
- HTP.m: Hard Thresholding Pursuit
- MSA_Final.mat: Input signal
- MSA_Final_Results.mat: Results of estimation