ECN-619: Introduction To Compressed Sensing
Assignment-5
Due date: 8$^{\text{th}}$ April 2024, Total marks: 30

Note: This assignment will be graded, hence the solutions you submit must be your own.

In this assignment, you will implement and compare the Orthogonal Matching Pursuit (OMP), the Compressive Sampling Matching Pursuit Algorithms (CoSAMP), and the Hard Thresholding Pursuit (HTP) algorithms. Execute the following steps in Matlab and create a report (in MS-Word or pdf format) containing the plots as highlighted in blue color below.

1. (**20 marks**) Create a Matlab file *single_run.m*

   (a) For $N = 400, k = 10$, generate a random k-sparse vector x with random k-sized nonzero support and the nonzero coefficients distributed independently and identically as $\mathcal{N}(0, 1)$. **Plot x_i as a function of $i \in [N]$.**

   (b) Generate a real-valued measurement matrix A of size $m \times N$, where $m = 100$. The entries of A are generated as i.i.d. $\mathcal{N}(0, 1/m)$. Use the measurement matrix A to generate the m-dimensional noisy measurement vector $y = Ax + e$, where the entries of noise vector e are i.i.d. $\mathcal{N}(0, \sigma^2), \sigma = 0.1$.

   (c) Create a separate Matlab file *OMP.m* containing the function *OMP(..)* which implements the Orthogonal Matching Pursuit algorithm to reconstruct x from y. The *OMP* function should have the following input/output:

   Input: Noisy measurements y, measurement matrix A, true support size k.

   Output: x_est: OMP's estimate of the k-sparse ground truth x.

   (d) Call the Matlab function *OMP* implemented in step (c) to recover x as x_est, i.e.

   $$x\_est = OMP(y, A, k);$$

   Plot the ground truth x and its reconstruction x_est overlaid on top of each other in different colors, e.g.,

   ```
   figure;
   plot([1:N],x, 'b', 'linewidth', 2); hold on;
   plot([1:N],x_est, 'g', 'linewidth', 1);
   legend('ground truth','OMP output');
   ```

   (e) Create a separate matlab file *CoSAMP.m* containing the function *CoSAMP(..)* which implements the CoSAMP algorithm to reconstruct x from y. The *CoSAMP* function should have the following input/output:

   Input: Noisy measurements y, measurement matrix A, true support size k.

   Output: x_est: Estimate of the k-sparse ground truth x.

   (f) Call the Matlab function *CoSAMP* implemented in step (e) to recover x as x_est, i.e.

   $$x\_est = CoSAMP(y, A, k);$$

   Plot the ground truth x and its reconstruction x_est overlaid on top of each other in different colors, e.g.,

```
figure;
plot([1:N],x, 'b', 'linewidth', 2); hold on;
plot([1:N],x_est, 'g', 'linewidth', 1);
legend('ground truth','CoSAMP output');
```

(g) Create a separate matlab file *HTP.m* containing the function *HTP(..)* which implements the Hard Thresholding Pursuit (HTP) algorithm to reconstruct x from y. The *HTP* function should have the following input/output:

Input: Noisy measurements y, measurement matrix A, true support size k.

Output: x_est: Estimate of the k-sparse ground truth x.

(h) Call the Matlab function *HTP* implemented in step (e) to recover x as x_est, i.e.

$$x\_est = HTP(y, A, k);$$

Plot the ground truth x and its reconstruction x_est overlaid on top of each other in different colors, e.g.,

```
figure;
plot([1:N],x, 'b', 'linewidth', 2); hold on;
plot([1:N],x_est, 'g', 'linewidth', 1);
legend('ground truth','HTP output');
```

2. (**10 marks**) In this part, we compare the sparse recovery performances of OMP, CoSAMP and HTP-based reconstruction for different values of SNRs (Signal to Noise Ratios). Create a separate matlab file *regression.m* for this part.

   (a) Using *for-loop(s)*, run 50 independent sparse recovery trials for each of the following SNRs: $10, 15, 20, 25, 30$ dB, wherein for each SNR, the following steps have to be implemented.

      i. For $N = 500, k = 10$, generate an n-dimensional, k-sparse vector x with randomized k-sized nonzero support and the nonzero coefficients distributed independently and identically as $\mathcal{N}(0, 1)$.

      ii. Generate a real-valued measurement matrix A of size $m \times N$, where $m = 100$. The entries of A are generated as i.i.d. $\mathcal{N}(0, 1/m)$.

      iii. Use the measurement matrix A to generate the m-dimensional noisy measurement vector $y = Ax + e$, where the entries of noise vector e are i.i.d. $\mathcal{N}(0, \sigma^2)$. The noise variance σ^2 depends on the active SNR (in dB) as shown below.

$$SNR_{\text{linear}} = 10^{\left(\frac{SNR_{dB}}{10}\right)}$$

$$\sigma^2 = \frac{\text{Signal variance}}{SNR_{\text{linear}}} \qquad \text{(Signal variance = 1 in our case)}$$

      iv. Use *OMP*, *Co-Samp* and *HTP* Matlab functions from question-1 to reconstruct x from y as $xest\_OMP$, $xest\_CoSAMP$, and $xest\_HTP$ respectively.

      v. For each independent trial, compute the normalized mean-squared error (NMSE) and the normalized support error rate (NSER) for OMP-based recovery as defined below.

$$NMSE\_OMP = \frac{\|x - xest\_OMP\|_2^2}{\|x\|_2^2}$$

$$NSER\_OMP = \frac{|supp(x) \backslash supp(xest\_OMP)| + |supp(xest\_OMP) \backslash supp(x)|}{|supp(x)|}$$

Similarly, also compute $NMSE\_cosamp$, $NMSE\_HTP$, $NSER\_cosamp$ and $NSER\_HTP$, corresponding to CoSAMP and HTP-based reconstruction $xest\_cosamp$ and $xest\_HTP$, respectively. While computing the NSER, the set operator '\' is the set difference operation, i.e. $\mathcal{A}\backslash\mathcal{B} = \mathcal{A} \cap \mathcal{B}^C$.

For each SNR, compute the averaged values of $NMSE$ and $NSER$ for both OMP and CoSAMP-based recovery, averaged across the 50 independent trials.

(b) **Plot the averaged NMSE (in dB) as a function of SNR (in dB), for OMP, Co-SAMP and HTP-based recovery, in the same plot.**

(c) **Plot the averaged NSER (in dB) as a function of SNR (in dB), for OMP, Co-SAMP and HTP-based recovery, in the same plot.**

IMPORTANT: Please upload/submit on MS Teams a single zipped (compressed) folder containing all of the matlab scripts and the pdf report containing the required plots. Keep the compressed folder name as $A5\_ < student\_name > .zip$.