CS663 Assignment-5

Saksham Rathi, Kavya Gupta, Shravan Srinivasa Raghavan
Department of Computer Science,
Indian Institute of Technology Bombay

Question 4

Part (a)

We are given an $n \times n$ image $f(x, y)$ such that only $k \ll n^2$ elements in it are non-zero, where k is known and the locations of the non-zero elements are also known. We are also given a set of only m different Discrete Fourier Transform (DFT) coefficients of known frequencies, where $m < n^2$. We need to reconstruct the image from these m DFT coefficients.
Here is the 1D DFT matrix:

$$F = \begin{bmatrix} e^{-2j\pi(0)(0)/S} & e^{-2j\pi(0)(1)/S} & \cdots & e^{-2j\pi(0)(S-1)/S} \\ e^{-2j\pi(1)(0)/S} & e^{-2j\pi(1)(1)/S} & \cdots & e^{-2j\pi(1)(S-1)/S} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-2j\pi(S-1)(0)/S} & e^{-2j\pi(S-1)(1)/S} & \cdots & e^{-2j\pi(S-1)(S-1)/S} \end{bmatrix}$$

The 2D DFT of an image $f(x, y)$ is given by

$$F(u, v) = \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} f(x, y) e^{-j2\pi\left(\frac{ux}{n} + \frac{vy}{n}\right)} \tag{1}$$

We can write the above equation in matrix form as

$$F_{2D} = F_{1D} \otimes F_{1D} \tag{2}$$

where F_{2D} is the 2D DFT of the image, F_{1D} is the 1D DFT matrix, and $\otimes$ denotes the Kronecker product.
We can get the 2D DFT coefficients as follows:

$$Y = F_{2D} \cdot X \tag{3}$$

where Y is the 2D DFT coefficients, F_{2D} is the 2D DFT matrix, and X is the image (flattened version). Y has the size $n^2 \times 1$, F_{2D} has the size $n^2 \times n^2$, and X has the size $n^2 \times 1$.
We already know that only k elements in the vector X are non-zero. Moreover, the position of these non-zero values is already known. Also, every index in the vector X, will get multiplied by a corresponding column in the matrix F_{2D}. If that index is zero, we can ignore that particular column in the matrix F_{2D}. Therefore, we need to extract only those columns of F_{2D} which correspond to the non-zero elements in X. Let us denote this matrix as A. Let X_{nz} be the vector of non-zero elements of X. We can write Y as

$$Y = A \cdot X_{nz} \tag{4}$$

where A is of size $n^2 \times k$ and X_{nz} is of size $k \times 1$. Now, we also know that we are given only m elements of Y. Therefore, we can write Y as

$$Y' = B \cdot X_{nz} \tag{5}$$

where B is of size $m \times k$. (We have basically omitted all the rows of A which are not present in Y'). Now, we need to find X_{nz} from Y' and B. We can do this by minimizing the squared loss:

$$\min_{X_{nz}} ||Y' - B \cdot X_{nz}||_2^2 \tag{6}$$

Let $e^2 = ||Y' - B \cdot X_{nz}||_2^2 = (Y' - BX_{nz})^H(Y' - BX_{nz}) = ||Y'^2|| - 2X_{nz}^H B^H Y' + X_{nz}^H B^H B X_{nz}$, where H denotes the Hermitian transpose. We can minimize e^2 by taking the derivative with respect to X_{nz} and setting it to zero. We get

$$\frac{\partial e^2}{\partial X_{nz}} = -2B^H Y' + 2B^H B X_{nz} = 0 \tag{7}$$

Solving the above equation, we get

$$X_{nz} = (B^H B)^{-1} B^H Y' \tag{8}$$

This can be written in terms of pseudo-inverse as

$$X_{nz} = pinv(B)Y' \tag{9}$$

Now, we already know the positions of the non-zero elements in X. We can reconstruct the image by putting these values in the correct positions. Therefore, we have successfully reconstructed the image from the given m DFT coefficients.

Part (b)

We need to find the minimum value of m that our method will allow. We know that B is of size $m \times k$. This basically means that there are m equations (corresponding to the different DFT coefficients we have) and k unknowns (the non-zero elements of the image). For this system of equations to be solvable, we need $m \geq k$. Therefore, the minimum value of m that our method will allow is k.

Part (c)

No, our method will not work if k is known, but the locations of the non-zero elements are unknown. This is because, even if we get the zon-zero values of X from the DFT coefficients, we will not know where to put these values in the image.

A brute force way to solve this problem is to try all possible combinations of k non-zero elements in the image and check which combination gives the minimum error. So, in total there will be $\binom{n^2}{k}$ combinations and $k!$ permutations for each combination. Thereofore, the total number of possibilities will be $\binom{n^2}{k} \times k!$. This is actually a very large number even for a small image size. Therefore, this method is not feasible.

Another way to solve this problem is to use Orthogonal Matching Pursuit (OMP) algorithm. OMP is a greedy algorithm that tries to find the best k non-zero elements in the image. Here is the pseudo code of the algorithm:

Algorithm 1 Norm-zero based OMP reconstruction

Input:

- Measurement vector $\mathbf{y}$
- Measurement matrix $\mathbf{A}$
- Number of selected coefficients in each iteration r, by default $r = 1$
- Required precision ε

$\mathbb{K} \leftarrow \emptyset$
$\mathbf{e} \leftarrow \mathbf{y}$
while $\|\mathbf{e}\|_2 > \varepsilon$ **do**
    $(k_1, k_2, \ldots, k_r) \leftarrow$ positions of r highest values in $\mathbf{A}^H\mathbf{e}$
    $\mathbb{K} \leftarrow \mathbb{K} \cup \{k_1, k_2, \ldots, k_r\}$
    $\mathbf{A}_K \leftarrow$ columns of matrix $\mathbf{A}$ selected by set $\mathbb{K}$
    $\mathbf{X}_K \leftarrow \mathrm{pinv}(\mathbf{A}_K)\mathbf{y}$
    $\mathbf{y}_K \leftarrow \mathbf{A}_K\mathbf{X}_K$
    $\mathbf{e} \leftarrow \mathbf{y} - \mathbf{y}_K$
end while
$\mathbf{X} \leftarrow \begin{cases} \mathbf{0} & \text{for positions not in } \mathbb{K} \\ \mathbf{X}_K & \text{for positions in } \mathbb{K} \end{cases}$

Output:

- Reconstructed signal coefficients $\mathbf{X}$

Figure 1: Pseudo code of Orthogonal Matching Pursuit (OMP) algorithm