Maximum Likelihood Encoder and Fourier Transforms

February 19, 2019

Maximum Likelihood Encoder

A more in depth look at the example on page 36-37 in lecture 3.

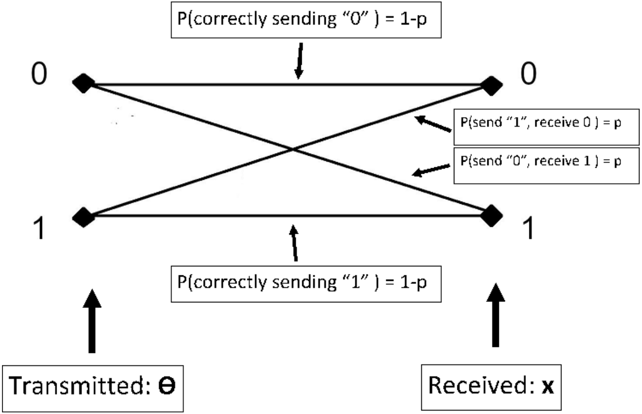

Figure 1: Diagram of the communications channel. The probability of sending the wrong message is p

Setting up the model

- The transmitted signal is given by $\theta[n] \in \{0, 1\}$, and denote the vector of bits sent as θ

Setting up the model

- The transmitted signal is given by $\theta[n] \in \{0, 1\}$, and denote the vector of bits sent as θ
- The channel noise $w[n] \in \{0, 1\}$ is represented by i.i.d. Bernoulli variables. The pmf of an individual $w[n]$ is

$$P(W = w[n]) = p^{w[n]}(1 - p)^{1 - w[n]}$$

Setting up the model

- The transmitted signal is given by $\theta[n] \in \{0, 1\}$, and denote the vector of bits sent as θ
- The channel noise $w[n] \in \{0, 1\}$ is represented by i.i.d. Bernoulli variables. The pmf of an individual $w[n]$ is

$$P(W = w[n]) = p^{w[n]}(1 - p)^{1 - w[n]}$$

- $w[n] = 1$ represents that an error has occurred in the channel transmission, because it always flips the transmitted signal.

Setting up the model

- The transmitted signal is given by $\theta[n] \in \{0, 1\}$, and denote the vector of bits sent as θ
- The channel noise $w[n] \in \{0, 1\}$ is represented by i.i.d. Bernoulli variables. The pmf of an individual $w[n]$ is

$$P(W = w[n]) = p^{w[n]}(1-p)^{1-w[n]}$$

- $w[n] = 1$ represents that an error has occurred in the channel transmission, because it always flips the transmitted signal.
- The received signal is

$$x[n] = \theta[n] \bigoplus w[n]$$

where $\bigoplus$ denotes addition modulo 2. Denote the vector containg the signal as $\mathbf{x}$, and the vector containing the noise as $\mathbf{w}$

Review of modulo addition

- Modulo 2 addition works by the formula $x \bigoplus y = \mathrm{remainder}(\frac{x+y}{2})$.
- If the addition results in an even number, the modulo is 0, and the modulo is 1 when the result is an odd number.
- In this example it is also equivalent to binary addition with no carry bit.

Review of modulo addition

- Modulo 2 addition works by the formula $x \bigoplus y = \mathrm{remainder}(\frac{x+y}{2})$.
- If the addition results in an even number, the modulo is 0, and the modulo is 1 when the result is an odd number.
- In this example it is also equivalent to binary addition with no carry bit.
- **Key Formula**: $x[n] = \theta[n] \bigoplus w[n] \rightarrow w[n] = x[n] \bigoplus \theta[n]$.

Review of modulo addition

- Modulo 2 addition works by the formula $x \bigoplus y = \mathrm{remainder}(\frac{x+y}{2})$.
- If the addition results in an even number, the modulo is 0, and the modulo is 1 when the result is an odd number.
- In this example it is also equivalent to binary addition with no carry bit.
- **Key Formula**: $x[n] = \theta[n] \bigoplus w[n] \rightarrow w[n] = x[n] \bigoplus \theta[n]$.
- To check this, take all four possible cases.

Review of modulo addition

- Modulo 2 addition works by the formula $x \bigoplus y = \mathrm{remainder}(\frac{x+y}{2})$.
- If the addition results in an even number, the modulo is 0, and the modulo is 1 when the result is an odd number.
- In this example it is also equivalent to binary addition with no carry bit.
- **Key Formula**: $x[n] = \theta[n] \bigoplus w[n] \rightarrow w[n] = x[n] \bigoplus \theta[n]$.
- To check this, take all four possible cases.
  1. $\theta[n] = 0,\ w[n] = 0$, $x[n] = \theta[n] \bigoplus w[n] = 0 \bigoplus 0 = 0 \rightarrow x[n] \bigoplus \theta[n] = 0 \bigoplus 0 = 0 = w[n]$

Review of modulo addition

- Modulo 2 addition works by the formula $x \bigoplus y = \mathrm{remainder}(\frac{x+y}{2})$.
- If the addition results in an even number, the modulo is 0, and the modulo is 1 when the result is an odd number.
- In this example it is also equivalent to binary addition with no carry bit.
- **Key Formula**: $x[n] = \theta[n] \bigoplus w[n] \to w[n] = x[n] \bigoplus \theta[n]$.
- To check this, take all four possible cases.
  1. $\theta[n] = 0,\ w[n] = 0\ , x[n] = \theta[n] \bigoplus w[n] = 0 \bigoplus 0 = 0 \to x[n] \bigoplus \theta[n] = 0 \bigoplus 0 = 0 = w[n]$
  2. $\theta[n] = 1,\ w[n] = 0\ , x[n] = \theta[n] \bigoplus w[n] = 1 \bigoplus 0 = 1 \to x[n] \bigoplus \theta[n] = 1 \bigoplus 1 = 0 = w[n]$

Review of modulo addition

- Modulo 2 addition works by the formula $x \bigoplus y = \text{remainder}(\frac{x+y}{2})$.
- If the addition results in an even number, the modulo is 0, and the modulo is 1 when the result is an odd number.
- In this example it is also equivalent to binary addition with no carry bit.
- **Key Formula**: $x[n] = \theta[n] \bigoplus w[n] \rightarrow w[n] = x[n] \bigoplus \theta[n]$.
- To check this, take all four possible cases.
  1. $\theta[n] = 0,\ w[n] = 0\ , x[n] = \theta[n] \bigoplus w[n] = 0 \bigoplus 0 = 0 \rightarrow x[n] \bigoplus \theta[n] = 0 \bigoplus 0 = 0 = w[n]$
  2. $\theta[n] = 1,\ w[n] = 0\ , x[n] = \theta[n] \bigoplus w[n] = 1 \bigoplus 0 = 1 \rightarrow x[n] \bigoplus \theta[n] = 1 \bigoplus 1 = 0 = w[n]$
  3. $\theta[n] = 0,\ w[n] = 1\ , x[n] = \theta[n] \bigoplus w[n] = 0 \bigoplus 1 = 1 \rightarrow x[n] \bigoplus \theta[n] = 1 \bigoplus 0 = 1 = w[n]$

Review of modulo addition

- Modulo 2 addition works by the formula $x \bigoplus y = \mathrm{remainder}(\frac{x+y}{2})$.
- If the addition results in an even number, the modulo is 0, and the modulo is 1 when the result is an odd number.
- In this example it is also equivalent to binary addition with no carry bit.
- **Key Formula**: $x[n] = \theta[n] \bigoplus w[n] \rightarrow w[n] = x[n] \bigoplus \theta[n]$.
- To check this, take all four possible cases.
  1. $\theta[n] = 0,\ w[n] = 0\ , x[n] = \theta[n] \bigoplus w[n] = 0 \bigoplus 0 = 0 \rightarrow x[n] \bigoplus \theta[n] = 0 \bigoplus 0 = 0 = w[n]$
  2. $\theta[n] = 1,\ w[n] = 0\ , x[n] = \theta[n] \bigoplus w[n] = 1 \bigoplus 0 = 1 \rightarrow x[n] \bigoplus \theta[n] = 1 \bigoplus 1 = 0 = w[n]$
  3. $\theta[n] = 0,\ w[n] = 1\ , x[n] = \theta[n] \bigoplus w[n] = 0 \bigoplus 1 = 1 \rightarrow x[n] \bigoplus \theta[n] = 1 \bigoplus 0 = 1 = w[n]$
  4. $\theta[n] = 1,\ w[n] = 1\ , x[n] = \theta[n] \bigoplus w[n] = 1 \bigoplus 1 = 0 \rightarrow x[n] \bigoplus \theta[n] = 0 \bigoplus 1 = 0 = w[n]$

Creating the likelihood function

- $p(\mathbf{x};\theta) = P(\mathbf{X} = \mathbf{x}) = P(\theta \bigoplus \mathbf{w} = \mathbf{x})$

$$= P(\mathbf{w} = \mathbf{x} \bigoplus \theta)$$

Creating the likelihood function

- $p(\mathbf{x};\theta) = P(\mathbf{X} = \mathbf{x}) = P(\theta \bigoplus \mathbf{w} = \mathbf{x})$

$$= P(\mathbf{w} = \mathbf{x} \bigoplus \theta)$$

- But $\mathbf{w}$ is a vector of i.i.d. Bernoulli random variables, so we know the pmf.

Creating the likelihood function

- $p(\mathbf{x};\theta) = P(\mathbf{X} = \mathbf{x}) = P(\theta \bigoplus \mathbf{w} = \mathbf{x})$
$$= P(\mathbf{w} = \mathbf{x} \bigoplus \theta)$$
- But $\mathbf{w}$ is a vector of i.i.d. Bernoulli random variables, so we know the pmf.
- $$P(\mathbf{w} = \mathbf{x} \bigoplus \theta) = \Pi_{n=0}^{N-1} p^{w[n]} (1-p)^{1-w[n]} =$$
$$p^{\sum_{n=0}^{N-1} x[n] \bigoplus \theta[n]} (1-p)^{N - \sum_{n=0}^{N-1} x[n] \bigoplus \theta[n]}$$

Creating the likelihood function

- $p(\mathbf{x};\theta) = P(\mathbf{X} = \mathbf{x}) = P(\theta \bigoplus \mathbf{w} = \mathbf{x})$
$$= P(\mathbf{w} = \mathbf{x} \bigoplus \theta)$$
- But $\mathbf{w}$ is a vector of i.i.d. Bernoulli random variables, so we know the pmf.
-
$$P(\mathbf{w} = \mathbf{x} \bigoplus \theta) = \Pi_{n=0}^{N-1} p^{w[n]}(1-p)^{1-w[n]} =$$
$$p^{\sum_{n=0}^{N-1} x[n] \bigoplus \theta[n]}(1-p)^{N-\sum_{n=0}^{N-1} x[n] \bigoplus \theta[n]}$$
- Now pull out the parts depending on $\theta[n]$:
$$P(\mathbf{W} = \mathbf{x} \bigoplus \theta) = (1-p)^N (\frac{p}{1-p})^{\sum_{n=0}^{N-1} x[n] \bigoplus \theta[n]}$$

Maximizing the likelihood function

- In communications systems, $p < .5$.
- This implies that $(\frac{p}{1-p}) < 1$.
- To maximize this term, minimize $\sum_{n=0}^{N-1} x[n] \bigoplus \theta[n]$.
- This term is called the "Hamming Distance".
- The next problem is to find a sequence $\theta[n]$ that minimizes the Hamming Distance.

Examples of Error Correcting Codes

- Repeating Code: Given a message, send the message 3 times, and vote on each bit e.g.:

$$\theta = [1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1]$$

- Most of the time, the correct bit will be displayed in 2/3 or 3/3 repetitions, but sometimes this will be wrong.
- Parity bit: Add a single bit to the signal that makes the number of 1's even, e.g. for 1-bit communication:

$$\theta[n] \in \{01, 11\}$$

- And for 2-bit:

$$\theta[n] \in \{000, 011, 101, 110\}$$

where the parity bits are represented in blue.

- This tells you whether you have an odd or even number of errors, and in the case of 1-bit communications corrects errors with high probability.

Comments on the Hamming Distance

- Named for Richard Hamming, who invented the formula in order to construct an error correcting coding system.
- Using the Hamming distance, errors in 2-bit communications can be detected. Errors in 1 bit communications can even be identified and corrected.
- Used to compare strings in text analysis to compare words of the same length.
- Used to compare gene codes in biology.

Fourier Transform - Kay example 4.2

- In many applications, the measurments of interest are modeled with a Fourier Series. For example, cell phones estimate the Fourier coefficients of your voice for 10-20 frequencies, and then send those numbers to the cell tower and out into the world.
- In this case the signal model is given as

$$x[n] = \sum_{k=1}^{M} a_k \cos(\frac{2\pi kn}{N}) + b_k \sin(\frac{2\pi kn}{N}) + w[n]$$

Fourier Transform - Kay example 4.2

- In many applications, the measurments of interest are modeled with a Fourier Series. For example, cell phones estimate the Fourier coefficients of your voice for 10-20 frequencies, and then send those numbers to the cell tower and out into the world.
- In this case the signal model is given as

$$x[n] = \sum_{k=1}^{M} a_k \cos(\frac{2\pi kn}{N}) + b_k \sin(\frac{2\pi kn}{N}) + w[n]$$

- Here $w[n] \sim N(0, \sigma^2)$ are the usual i.i.d. samples of white gaussian noise.

Fourier Transform - Kay example 4.2

- In many applications, the measurments of interest are modeled with a Fourier Series. For example, cell phones estimate the Fourier coefficients of your voice for 10-20 frequencies, and then send those numbers to the cell tower and out into the world.
- In this case the signal model is given as

$$x[n] = \sum_{k=1}^{M} a_k \cos(\frac{2\pi kn}{N}) + b_k \sin(\frac{2\pi kn}{N}) + w[n]$$

- Here $w[n] \sim N(0, \sigma^2)$ are the usual i.i.d. samples of white gaussian noise.
- Denote $\mathbf{x} = \begin{bmatrix} x[0], & x[1], & \dots & x[n] \end{bmatrix}^T$.

Fourier Transform - Kay example 4.2

- In many applications, the measurments of interest are modeled with a Fourier Series. For example, cell phones estimate the Fourier coefficients of your voice for 10-20 frequencies, and then send those numbers to the cell tower and out into the world.
- In this case the signal model is given as

$$x[n] = \sum_{k=1}^{M} a_k \cos(\frac{2\pi kn}{N}) + b_k \sin(\frac{2\pi kn}{N}) + w[n]$$

- Here $w[n] \sim N(0, \sigma^2)$ are the usual i.i.d. samples of white gaussian noise.
- Denote $\mathbf{x} = \begin{bmatrix} x[0], & x[1], & \dots & x[n] \end{bmatrix}^T$.
- The vector of parameters to be estimated is $\theta = \begin{bmatrix} a_1, & a_2, & \dots & a_M, & b_1, & b_2 & \dots & b_M \end{bmatrix}^T$

Linear Model Formulation

- Define the model matrix $\mathbf{H}$ as:

$$\begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \\ \cos(\frac{2\pi}{N}) & \cdots & \cos(\frac{2\pi M}{N}) & \sin(\frac{2\pi}{N}) & \cdots & \sin(\frac{2\pi M}{N}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \cos(\frac{2\pi(N-1)}{N}) & \cdots & \cos(\frac{2\pi M(N-1)}{N}) & \sin(\frac{2\pi(N_1)}{N}) & \cdots & \sin(\frac{2\pi M(N-1)}{N}) \end{bmatrix}$$

Linear Model Formulation

- Define the model matrix $\mathbf{H}$ as:

$$\begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \\ \cos(\frac{2\pi}{N}) & \cdots & \cos(\frac{2\pi M}{N}) & \sin(\frac{2\pi}{N}) & \cdots & \sin(\frac{2\pi M}{N}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \cos(\frac{2\pi(N-1)}{N}) & \cdots & \cos(\frac{2\pi M(N-1)}{N}) & \sin(\frac{2\pi(N_1)}{N}) & \cdots & \sin(\frac{2\pi M(N-1)}{N}) \end{bmatrix}$$

- The columns of $H = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \ldots & \mathbf{h}_{2M} \end{bmatrix}$ are given by either $\cos(\frac{2\pi in}{N})$ or $\sin(\frac{2\pi in}{N})$ for $i = 1, 2, \ldots, M$ and $n = 0, 1, \ldots, N-1$

Linear Model Formulation

- Define the model matrix $\mathbf{H}$ as:

$$\begin{bmatrix} 1 & \cdots & 1 & 0 & \cdots & 0 \\ \cos(\frac{2\pi}{N}) & \cdots & \cos(\frac{2\pi M}{N}) & \sin(\frac{2\pi}{N}) & \cdots & \sin(\frac{2\pi M}{N}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \cos(\frac{2\pi(N-1)}{N}) & \cdots & \cos(\frac{2\pi M(N-1)}{N}) & \sin(\frac{2\pi(N_1)}{N}) & \cdots & \sin(\frac{2\pi M(N-1)}{N}) \end{bmatrix}$$

- The columns of $H = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 & \ldots & \mathbf{h}_{2M} \end{bmatrix}$ are given by either $\cos(\frac{2\pi in}{N})$ or $\sin(\frac{2\pi in}{N})$ for $i = 1, 2, \ldots, M$ and $n = 0, 1, \ldots, N-1$
- The linear model form of the fourier series estimation problem is given by

$$\mathbf{x} = \mathbf{H}\theta + \mathbf{w}$$

The Information Matrix

- Recall that the Fisher Information Matrix is given by $\frac{1}{\sigma^2}\mathbf{H}^T\mathbf{H}$ for linear models.
- $\mathbf{H}^T\mathbf{H} = \begin{bmatrix} \mathbf{h}_1^T \\ \vdots \\ \mathbf{h}_{2M}^T \end{bmatrix} \begin{bmatrix} \mathbf{h}_1 & \ldots & \mathbf{h}_{2M} \end{bmatrix} =$

$$\begin{bmatrix} \mathbf{h_1^t h_1} & \mathbf{h_1^T h_2} & \cdots & \mathbf{h_1^T h_{2M}} \\ \mathbf{h_2^t h_1} & \mathbf{h_2^T h_2} & \cdots & \mathbf{h_2^T h_{2M}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h_{2M}^t h_1} & \mathbf{h_{2M}^T h_2} & \cdots & \mathbf{h_{2M}^T h_{2M}} \end{bmatrix}$$

- There are 3 possible cases here.

Case 1

- Case 1: $\mathbf{h}_i^T \mathbf{h}_j = \sum_{n=0}^{N-1} \cos(\frac{2\pi i n}{N}) \cos(\frac{2\pi j n}{N})$

Case 1

- Case 1: $\mathbf{h}_i^T \mathbf{h}_j = \sum_{n=0}^{N-1} \cos(\frac{2\pi i n}{N}) \cos(\frac{2\pi j n}{N})$
- If $i = j$, then:

$$\sum_{n=0}^{N-1} \cos(\frac{2\pi i n}{N})^2 = \sum_{n=0}^{N=N-1} \frac{1}{2} + \frac{1}{2} \cos(\frac{4\pi i n}{N}) =$$

$$\frac{N}{2} + \frac{1}{2} \sum_{n=0}^{N-1} \cos(\frac{4\pi i n}{N}) = \frac{N}{2} + 0$$

Case 1

- Case 1: $\mathbf{h}_i^T\mathbf{h}_j = \sum_{n=0}^{N-1} \cos(\frac{2\pi in}{N})\cos(\frac{2\pi jn}{N})$
- If $i = j$, then:

$$\sum_{n=0}^{N-1} \cos(\frac{2\pi in}{N})^2 = \sum_{n=0}^{N=N-1} \frac{1}{2} + \frac{1}{2}\cos(\frac{4\pi in}{N}) =$$

$$\frac{N}{2} + \frac{1}{2}\sum_{n=0}^{N-1} \cos(\frac{4\pi in}{N}) = \frac{N}{2} + 0$$

- For a proof of the final sum, use "Lagrange's Trigonometric Identities". The proof of these identities expands the functions in terms of complex exponentials.

Case 1

- Case 1: $\mathbf{h}_i^T \mathbf{h}_j = \sum_{n=0}^{N-1} \cos(\frac{2\pi in}{N}) \cos(\frac{2\pi jn}{N})$
- If $i = j$, then:

$$\sum_{n=0}^{N-1} \cos(\frac{2\pi in}{N})^2 = \sum_{n=0}^{N=N-1} \frac{1}{2} + \frac{1}{2} \cos(\frac{4\pi in}{N}) =$$

$$\frac{N}{2} + \frac{1}{2} \sum_{n=0}^{N-1} \cos(\frac{4\pi in}{N}) = \frac{N}{2} + 0$$

- For a proof of the final sum, use "Lagrange's Trigonometric Identities". The proof of these identities expands the functions in terms of complex exponentials.
- If $i \neq j$ then:

$$\sum_{n=0}^{N-1} \cos(\frac{2\pi in}{N}) \cos(\frac{2\pi jn}{N}) = \frac{1}{2} \sum_{n=0}^{N-1} \cos(\frac{2\pi(i+j)n}{N}) + \cos(\frac{2\pi(i-j)n}{N}) = 0$$

using the same identity as before for the finite sum of cosines.

Case 2 and 3

- Case 2: $\mathbf{h}_i^T \mathbf{h}_j = \sum_{n=0}^{N-1} \sin(\frac{2\pi i n}{N}) \sin(\frac{2\pi j n}{N})$

Case 2 and 3

- Case 2: $\mathbf{h}_i^T\mathbf{h}_j = \sum_{n=0}^{N-1} \sin(\frac{2\pi in}{N}) \sin(\frac{2\pi jn}{N})$
- This case is almost identical with the last case.

Case 2 and 3

- Case 2: $\mathbf{h}_i^T\mathbf{h}_j = \sum_{n=0}^{N-1} \sin(\frac{2\pi in}{N})\sin(\frac{2\pi jn}{N})$
- This case is almost identical with the last case.
- $\mathbf{h}_i^T\mathbf{h}_j = \begin{cases} \frac{N}{2} \text{ for i} = \text{j} \\ 0 \text{ else} \end{cases}$

Case 2 and 3

- Case 2: $\mathbf{h}_i^T\mathbf{h}_j = \sum_{n=0}^{N-1} \sin(\frac{2\pi in}{N})\sin(\frac{2\pi jn}{N})$
- This case is almost identical with the last case.
- $\mathbf{h}_i^T\mathbf{h}_j = \begin{cases} \frac{N}{2} \text{ for i = j} \\ 0 \text{ else} \end{cases}$
- Case 3 is the sum $\mathbf{h}_i^T\mathbf{h}_j = \sum_{n=0}^{N-1} \cos(\frac{2\pi in}{N})\sin(\frac{2\pi jn}{N})$

Case 2 and 3

- Case 2: $\mathbf{h}_i^T \mathbf{h}_j = \sum_{n=0}^{N-1} \sin(\frac{2\pi i n}{N}) \sin(\frac{2\pi j n}{N})$
- This case is almost identical with the last case.
- $\mathbf{h}_i^T \mathbf{h}_j = \begin{cases} \frac{N}{2} \text{ for i = j} \\ 0 \text{ else} \end{cases}$
- Case 3 is the sum $\mathbf{h}_i^T \mathbf{h}_j = \sum_{n=0}^{N-1} \cos(\frac{2\pi i n}{N}) \sin(\frac{2\pi j n}{N})$
- This reduces to a similar expression as case 1.

$$\frac{1}{2} \sum_{n=0}^{N-1} \sin(\frac{2\pi(i+j)n}{N}) - \sin(\frac{2\pi(i-j)n}{N}) = 0$$

MVU Estimator

- Using trig identities, it's easy to see that $\mathbf{H}^T\mathbf{H} = \frac{N}{2}\mathbb{I}$, where $\mathbb{I}$ is the identity matrix.
- Then the optimal least squares estimator of the fourier coefficients is

$$\hat{\mathbf{\Theta}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{x} = \frac{2}{N}\begin{bmatrix} \mathbf{h}_1^T\mathbf{x} \\ \mathbf{h}_2^T\mathbf{x} \\ \vdots \\ \mathbf{h}_{2M}^T\mathbf{x} \end{bmatrix}$$

- The specific coefficients are

$$\hat{a}_k = \frac{2}{N}\sum_{n=0}^{N-1} x[n]\cos(\frac{2\pi kn}{N})$$

$$\hat{b}_k = \frac{2}{N}\sum_{n=0}^{N-1} x[n]\sin(\frac{2\pi kn}{N})$$

- Congratulations, we have re-invented the Discrete Fourier Transform!

Comments on the Fourier Transform

- Fourier series/transforms are ubiquitous in mathematics and engineering.
- Trigonometric functions are not the only available basis functions, for example the orthogonal polynomials are often used as bases to solve chaotic dynamical systems.
- This is a good example of data compression.
- Convolution (seen in the Systems Identification slides) operation is turned into multiplication in the fourier domain.
- A Fourier transform is the only thing standing between encryption systems and total global chaos (this is a slight exaggeration).