MAT 167: Applied Linear Algebra

Note Title

LECTURE 01

Course Objectives:

* To learn the importance of linear algebra in practical problems and applications, in particular,
  - machine learning & pattern recognition
  - data mining & search engines
  - signal & image processing

* To learn important & useful concepts of linear algebra, e.g.,
  - linear transformations
  - bases & orthogonality
  - projections & least squares method
  - various matrix decompositions
    LU, QR, eigenvalue, and
    SVD (singular value decomposition)

* To enhance your understanding of the above concepts & applications through the use of ~~MATLAB~~ Julia

Motivation: Vector/Matrix Representation of Datasets

Example 1. Music Signals and Signal Compression
⇒ ~~MATLAB~~ Julia Demo!

Example 2. Face Image Database
⇨ Figures

Example 3. Term-Document Matrices for Search

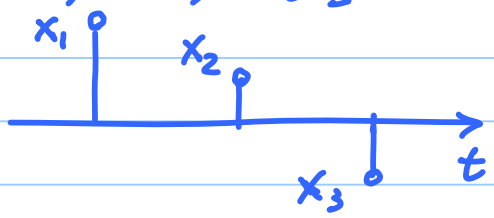
Example 4. Link Graphs of Webpages

Example 1: Music Signals & Signal Compression

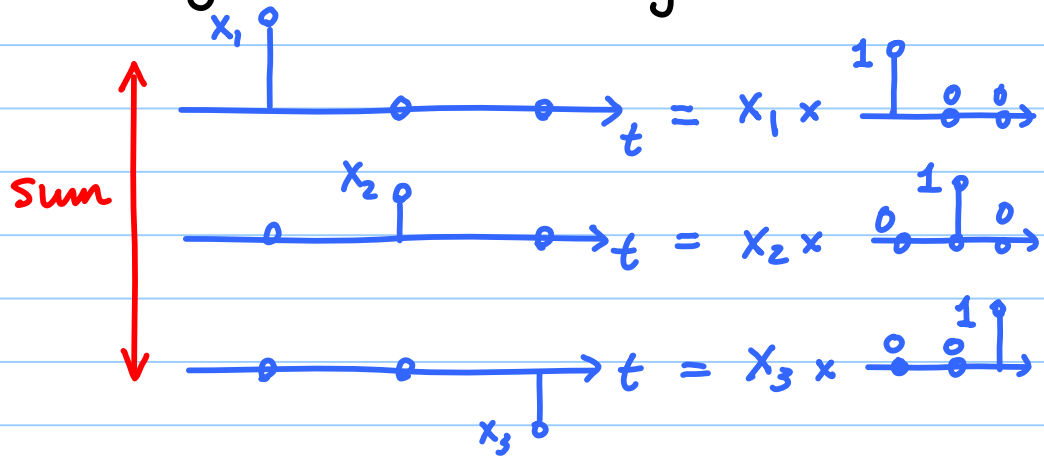
Consider a very short signal consisting of 3 samples

actual music signal may have 10^6 samples or more

$\mathbb{X} = [x_1, x_2, x_3]^T = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$



This vector can be represented exactly as the following sum



In other words

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = x_1 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + x_2 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + x_3 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\Leftrightarrow \quad \mathbb{X} = x_1 \mathbb{e}_1 + x_2 \mathbb{e}_2 + x_3 \mathbb{e}_3$$

Any $\mathbb{X} \in \mathbb{R}^3$ can be written exactly using the linear combination of $\mathbb{e}_1, \mathbb{e}_2, \mathbb{e}_3$

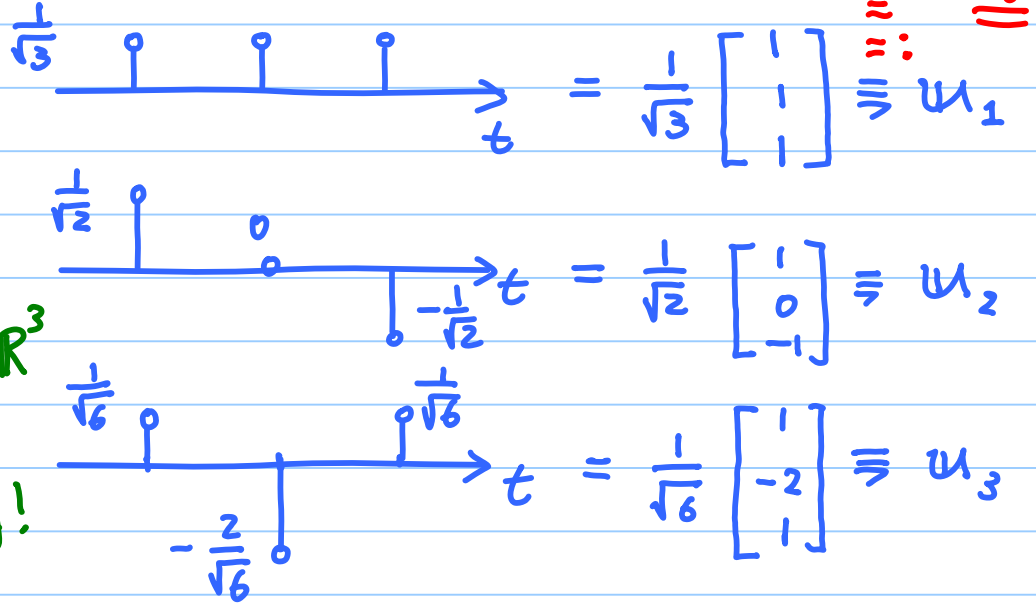
This can also be written as

$$x = \left[\, e_1 \,\middle|\, e_2 \,\middle|\, e_3 \,\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

basis vectors ← basis matrix

However, depending on a signal, there may be more efficient way to represent / approximate a given vector x.

e.g., consider the following vectors instead of e_1, e_2, e_3.



any $x \in \mathbb{R}^3$ can be written this way!

$$x = a_1 u_1 + a_2 u_2 + a_3 u_3$$

Given x, how to compute a_1, a_2, a_3?

⇒ We'll learn how when we discuss "orthogonality"!

We can also write this as

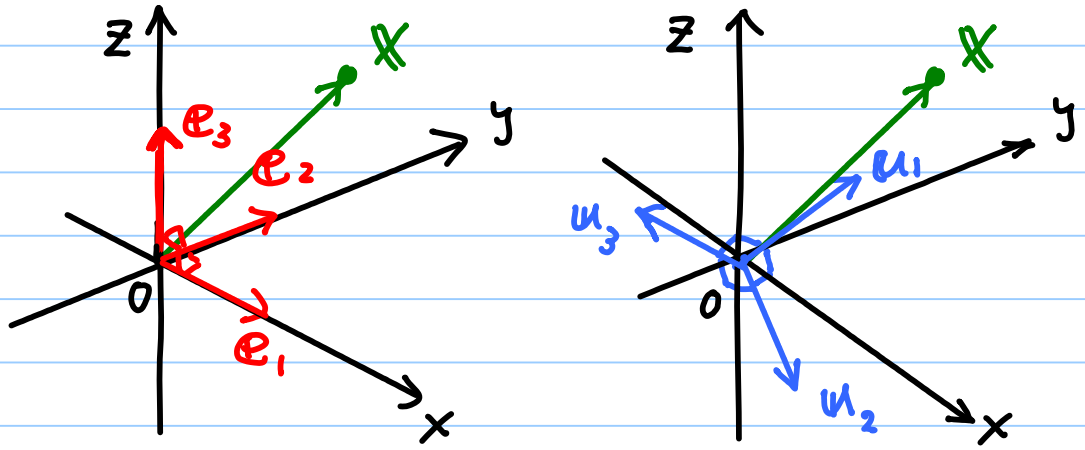
$$X = \left[\, u_1 \,\middle|\, u_2 \,\middle|\, u_3 \,\right] \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & -\frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

basis matrix

coordinate vector of X relative to U

$$= U a \Rightarrow a = U^T X$$

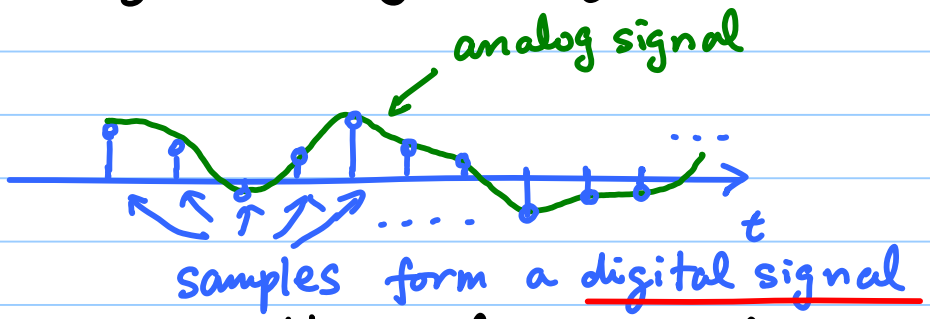
Note



But, we can only visualize such orthogonality up to 3D. The orthogonality can be generalized to any dimension $\mathbb{R}^n$, $n \geq 1$

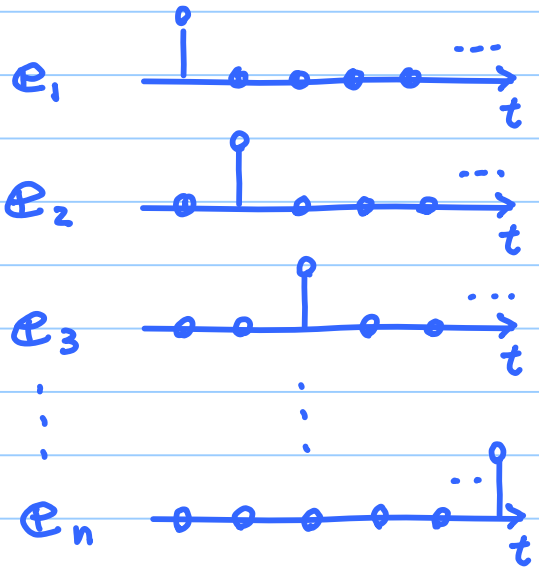
For a real music signal with n: huge, we can proceed similarly!

Input signal (say from your mp3 file).



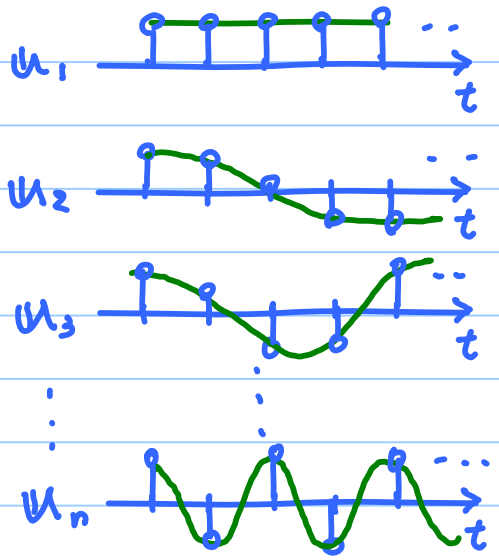
Again, we can write such a digital signal $X = [x_1, x_2, \cdots, x_n]^T$ as a linear combination of basis vectors.

The basis I



$$X = x_1 e_1 + \cdots + x_n e_n$$

The basis II



$$X = a_1 u_1 + \cdots + a_n u_n$$

Appropriately choosing the basis $\{u_1, \cdots, u_n\}$ we can Compress (or more precisely, compactly approximate) the input signal X!

For example, let's use the so-called **Discrete Cosine Transform (DCT)** basis for $\{u_1, \cdots, u_n\}$.
Then do the following operation:

(1) Compute the coefficient vector
$a = [a_1, \cdots, a_n]^T$.

(2) For $j = 1 : n$,
$$\tilde{a}_j := \begin{cases} a_j & \text{if } |a_j| \geq \theta \text{ (a theshold)} \\ 0 & \text{otherwise} \end{cases}$$

(3) Let $\tilde{a} := [\tilde{a}_1, \cdots, \tilde{a}_n]^T$, and reconstruct an approximate signal
$$\tilde{x} = U\tilde{a} = \tilde{a}_1 u_1 + \cdots + \tilde{a}_n u_n$$

Julia
$\Rightarrow$ ~~MATLAB~~ Demo with my music signal $n = 800{,}791$

- $\theta = 0.1 \Rightarrow$ 0.5% of coeff's survived
- $\theta = 0.01 \Rightarrow$ 6% of coeff's survived

Also interesting to hear the approximation error $x - \tilde{x}$!

If you do the same experiment using $\{e_1, \cdots, e_n\}$ instead of $\{u_1, \cdots, u_n\}$, you'll hear a huge difference!
That is, for a usual music signal, the DCT basis is better than the standard (or canonical) basis $\{e_1, \cdots e_n\}$.