MAT 167: Applied Linear Algebra

Note Title

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

Motivation: Vector/Matrix Representation of Datasets

<u>Example 1</u>. Music Signals and Signal Compression

$\Rightarrow$ MATLAB Demo!

<u>Example 2</u>. Face Image Database

$\Rightarrow$ Figures

<u>Example 3</u>. Term-Document Matrices for Search

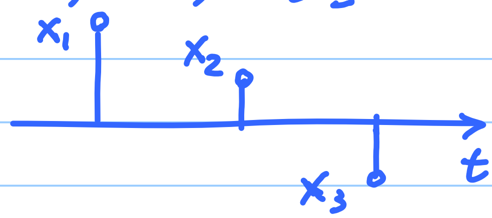
<u>Example 4</u>. Link Graphs of Webpages

Example 1: Music Signals & Signal Compression

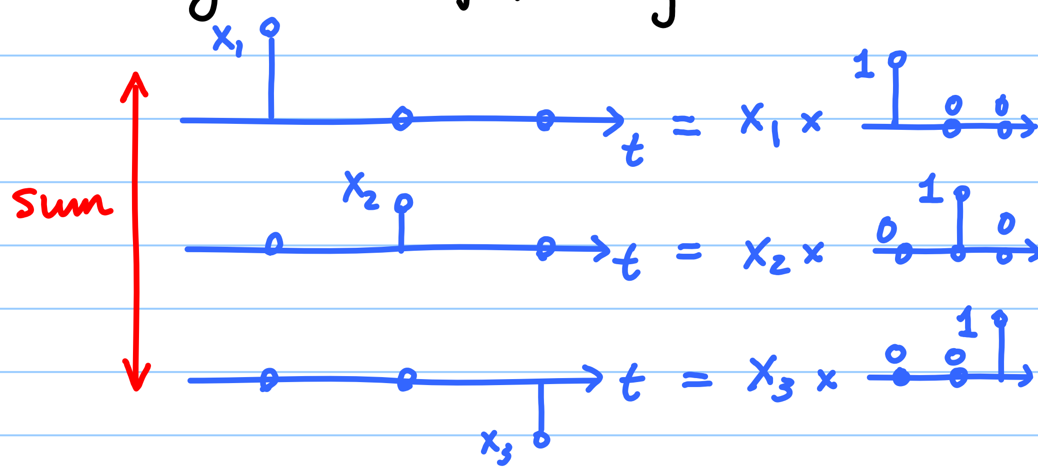
Consider a very short signal consisting of 3 samples

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 $\mathbb{e}_1, \mathbb{e}_2, \mathbb{e}_3$

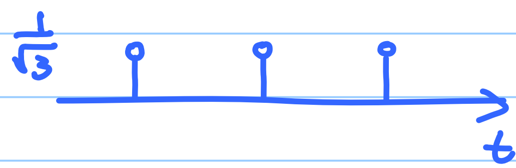
This can also be written as

$$\mathbb{x} = \underbrace{\begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix}}_{\text{basis matrix}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{basis matrix}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

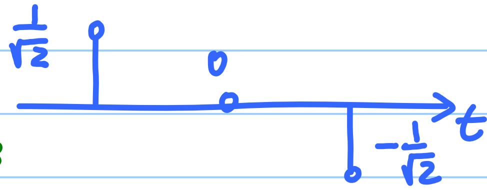
basis vectors

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

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

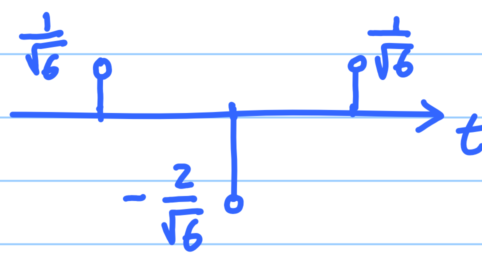


$$= \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \overset{\triangle}{\equiv} u_1 \quad \text{def}$$



$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \equiv u_2$$

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



$$= \frac{1}{\sqrt{6}} \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \equiv u_3$$

$$\mathbb{x} = a_1 u_1 + a_2 u_2 + a_3 u_3$$

Given $\mathbb{x}$, how to compute a_1, a_2, a_3?

$\Rightarrow$ We'll learn how when we discuss "orthogonality"!

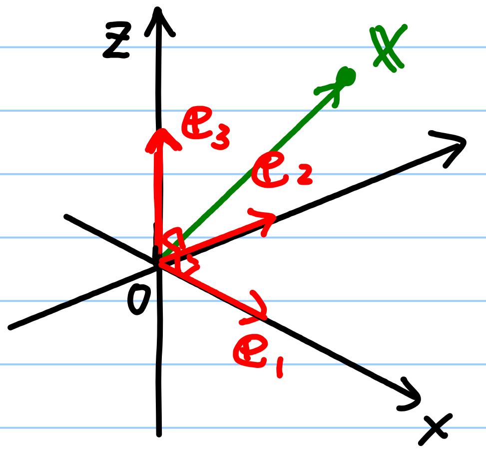
We can also write this as

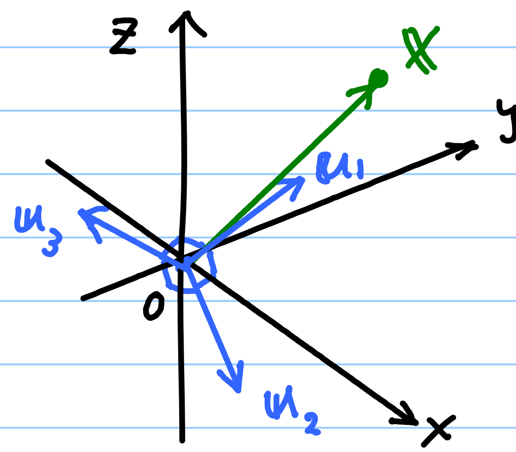
$$\mathbf{x} = \underbrace{\begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}}_{\text{basis matrix}} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & 0 & -\frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{6}} \end{bmatrix}}_{\text{basis matrix}} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

$$= U a \Rightarrow a = U^T \mathbf{x}$$

coordinate vector of $\mathbf{x}$ relative to U

Note

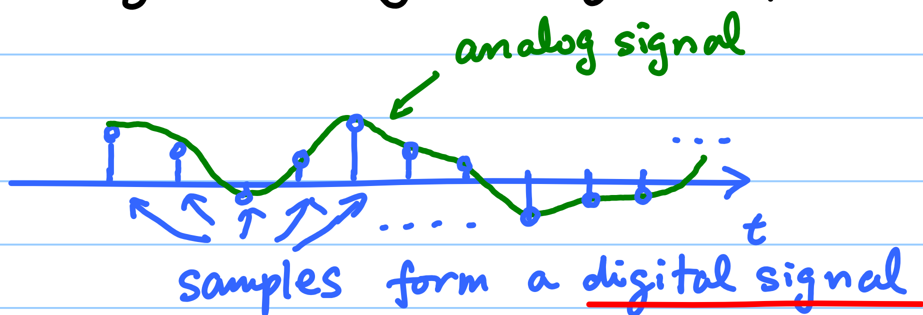




But, we can only visualize such orthogonality up to 3D. The orthogonality can be generalized to any dimension $\mathbb{R}^n$, $n \geq 1$

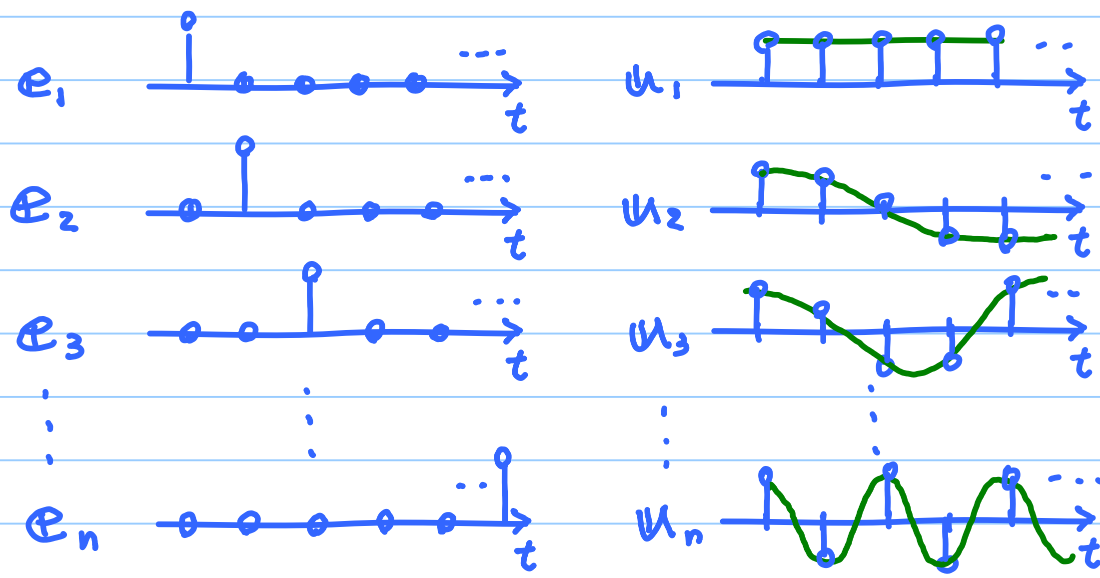
For a real music signal with n: huge, we can proceed similarly!

Input signal (say from your mp3 file).



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

The basis I | The basis II



$$\Downarrow \qquad\qquad\qquad \Downarrow$$

$$\mathbb{x} = x_1 \mathbb{e}_1 + \cdots + x_n \mathbb{e}_n \qquad \mathbb{x} = a_1 \mathbb{u}_1 + \cdots + a_n \mathbb{u}_n$$

Appropriately choosing the basis $\{\mathbb{u}_1, \cdots, \mathbb{u}_n\}$ we can compress (or more precisely, compactly approximate) the input signal $\mathbb{x}$!

For example, let's use the so-called Discrete Cosine Transform (DCT) basis for $\{\mathbf{u}_1, \cdots, \mathbf{u}_n\}$.

Then do the following operation:

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

(2) For $j = 1:n$,

$$\tilde{a}_j := \begin{cases} a_j & \text{if } |a_j| \geq \theta \text{ (a threshold)} \\ 0 & \text{otherwise} \end{cases}$$

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

$$\tilde{\mathbf{x}} = U\tilde{\mathbf{a}} = \tilde{a}_1\mathbf{u}_1 + \cdots + \tilde{a}_n\mathbf{u}_n$$

$n = 800{,}791$

$\Rightarrow$ MATLAB Demo with my music signal

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

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

If you do the same experiment using $\{\mathbf{e}_1, \cdots, \mathbf{e}_n\}$ instead of $\{\mathbf{u}_1, \cdots, \mathbf{u}_n\}$, you'll hear a huge difference!

That is, for a usual music signal, the DCT basis is better than the standard (or canonical) basis $\{\mathbf{e}_1, \cdots, \mathbf{e}_n\}$.