Koneru Lakshmaiah Education Foundation
(Category -1, Deemed to be University estd. u/s. 3 of the UGC Act, 1956)
❖ Approved by AICTE ❖ ISO 21001:2018 Certified
Campus: Green Fields, Vaddeswaram - 522 302, Guntur District, Andhra Pradesh, INDIA.
Phone No. +91 8645 - 350 200; www.klef.ac.in; www.klef.edu.in; www.kluniversity.in
Admin Off: 29-36-38, Museum Road, Governorpet, Vijayawada - 520 002. Ph: +91 - 866 - 3500122, 2576129

Course Name: Mathematics for AI
Explain course handout:
Session-1: Understand vector representations and similarity measures
Vectors

- Definition of a vector in $\mathbb{R}^n$

  A **vector in** $\mathbb{R}^n$ is an **ordered n-tuple of real numbers**.

$$\mathbf{x} = (x_1, x_2, x_3, \dots, x_n), \text{where } x_i \in \mathbb{R}$$

  Here:

- n is a **positive integer** called the **dimension**,
- each x_i is called a **component (or coordinate)** of the vector.

  Set-Theoretic Description

$$\mathbb{R}^n = \{(x_1, x_2, \dots, x_n) \mid x_i \in \mathbb{R}\}$$

  Geometric Interpretation

- In $\mathbb{R}^2$: a vector represents a point or directed line segment in a plane.
- In $\mathbb{R}^3$: a vector represents a point or directed line segment in space.
- For $n > 3$: vectors exist in **higher-dimensional spaces** (cannot be visualized directly but handled algebraically).

  Algebraic Interpretation

  A vector in $\mathbb{R}^n$ is an **element of a real vector space** equipped with:

- vector addition
- scalar multiplication

Example-1: $\mathrm{u} = (2,4) \in \mathbb{R}^2$ **and v** $= (2, -1, 3) \in \mathbb{R}^3$.

  Notation

- Row vector: $(x_1, x_2, \dots, x_n)$
- Column vector:

$$\begin{pmatrix} x_1 \\ x_2 \\ \\ x_n \end{pmatrix}$$

Vector addition in $\mathbb{R}^{\boldsymbol{n}}$: Let $u = (x_1, x_2, \dots, x_n)$ and v= $(y_1, y_2, \dots, y_n)$ be two vectors in $\mathbb{R}^{\boldsymbol{n}}$ then, their sum $u + v$ is defined as:

$$u + v = (x_1, x_2, \dots, x_n) + (y_1, y_2, \dots, y_n) = (x_1 + y_1, x_2 + y_2, \dots, x_n + y_n).$$

Example 2: Let $\boldsymbol{v_1} = (2, -1,3) \in \mathbb{R}^3$ and $\boldsymbol{v_2} = (3, -4,6) \in \mathbb{R}^3$ then Let $\boldsymbol{v_1} + \boldsymbol{v_2} = (5, -5,9) \in \mathbb{R}^3$.

Example 3: A drone flies 3 km east and then 4 km north. Find the resultant displacement.

Example 4: Three forces **5 N along x-axis**, **–3 N along y-axis**, and **4 N along z-axis** act on a particle. Find the **resultant force**.

Dot Product

$$\mathrm{u} \cdot v = \sum_{i=1}^{n} x_i \, y_i$$

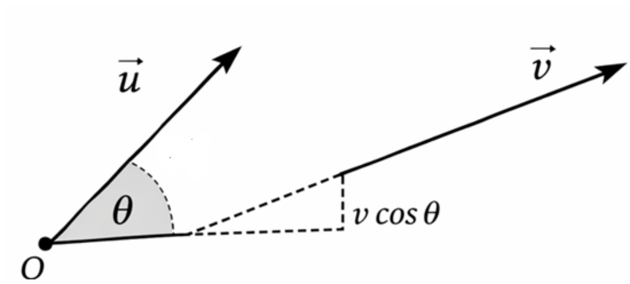
Problem 1: Let $\mathrm{u} = (1, -2,3,4), \mathrm{v} = (2,0, -1,5)$. Compute $\mathrm{u} \cdot \mathrm{v}$.

Problem 2: A force $\vec{F} = (3, -2,6)$ N moves a body through a displacement $\vec{s} = (4,1,2)$ m. Compute the work done.

Properties:

(i) Commutative: $\mathbf{u} \cdot \mathbf{v} = \mathbf{v} \cdot \mathbf{u}$ for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$.
(ii) Distributive: $(\mathbf{u} + \mathbf{v}) \cdot \mathbf{w} = \mathbf{u} \cdot \mathbf{w} + \mathbf{v} \cdot \mathbf{w}$.

Geometric meaning: $u \cdot \mathbf{v} = \| \mathbf{u} \| \, . \, \| \mathbf{v} \| \cos \theta$

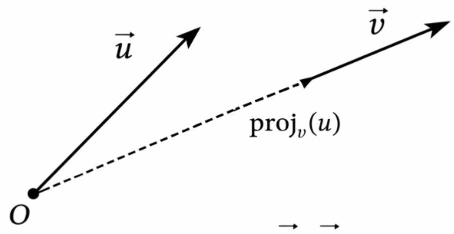


Vector Norms: Euclidean norm: $\| \mathbf{v} \| = \sqrt{\sum y_i^2}$

Meaning of magnitude and length: Distance between vectors

Problem-1: Find the Euclidean norm of $\mathbf{v} = (1, \; -2, \; 2, \; 1)$.
Problem-2: Let $\boldsymbol{v} = (x, \; 2x, \; 3x)$, Find $\| \boldsymbol{v} \|$in terms of x.

Projection of One Vector onto Another: The projection of a vector u onto another vector v is the vector component of uthat lies in the direction of v. It is given by



$$\text{proj}_v(u) = \frac{\vec{u}\cdot\vec{v}}{||\vec{v}||^2}\,\vec{v}$$

Problem-1: Find the projection of $u = (3,4)$ onto $v = (1,0)$.
Problem-2: Show that the vector $u - \text{proj}_{\,v}(u)$ is orthogonal to v.

Problem-3: Let force $\mathbf{F} = (5,3,4)$ acts on a body which moves along the direction $\mathbf{d} = (1,2,2)$. Find the component of the force along the direction of motion.

Session-2: Represent AI data using vectors
Representing Vectors Digitally

A vector as a list/array: $x = [x_1, x_2, \dots, x_n]$

- Concept of dimension
- High-dimensional vectors (motivation for text data)

Example 1: Three feature embeddings are $\vec{e}_1 = (1,2,3)$, $\vec{e}_2 = (2,-1,1)$, $\vec{e}_3 = (-1,1,2)$. Find the combined embedding vector.
Solution: $\vec{R} = \vec{e}_1 + \vec{e}_2 + \vec{e}_3$
$= (1+2-1)\hat{\imath} + (2-1+1)\hat{\jmath} + (3+1+2)\hat{k} = (2, 2, 6)$
Magnitude: $|\vec{R}| = \sqrt{4+4+36} = \sqrt{44} \approx 6.63$
Answer: Combined embedding = **(2, 2, 6).**

Problem 2: Let $u = (x, 2x, 3x, \dots, nx), v = (1,1,1, \dots, 1)$. Find $u \cdot v$.

Problem 3: Wind velocity is $\vec{v} = (6,8,2)$ m/s. The runway direction is given by the unit vector $\hat{n} = \left(\frac{3}{5}, \frac{4}{5}, 0\right)$. Find the wind component along the runway.

Problem-4: A feature vector in an ML model is $\mathbf{x} = (4,\ 3,\ 12)$, Compute its Euclidean norm.
Problem-5: If $\|\mathbf{v}\| = 5$ and $\mathbf{v} = (a,\ b,\ c)$. find all possible values of a, b, c satisfying the condition.
Cosine Similarity (Core Concept)

Definition: $\text{Cosine Similarity(a, b)} = \frac{a\cdot b}{\|a\|\,\|b\|}$
Why cosine?

- Independent of magnitude
- Measures directional similarity
- Values lie in $[-1,1]$

Value	Meaning
1	Identical direction
0	Orthogonal (unrelated)
−1	Opposite direction

Case Study 1: Movie Recommendation Based on Plot Similarity 🎬
Problem
A streaming platform wants to **recommend movies with similar stories** based on their descriptions.

Movie	Description
D1	"A young hero saves the world from aliens"
D2	"A brave hero fights aliens to save Earth"
D3	"A romantic story about love and friendship"

Step 1: **Text Preprocessing**
We first clean the text:

- Convert to lowercase
- Remove stop words (a, the, to, from, etc.)
- Tokenize words

Key words extracted

- D1: {young, hero, saves, world, aliens}
- D2: {brave, hero, fights, aliens, save, earth}
- D3: {romantic, story, love, friendship}

Step 2: **Vocabulary Construction**
Unique words across all descriptions:
[young, hero, saves, world, aliens, brave, fights, earth, romantic, story, love, friendship]
Step 3: Vector Representation (Bag of Words)
Each movie is represented as a vector based on word presence (1 = present, 0 = absent).

Word ↓ / Movie →	young	hero	saves	world	aliens	brave	fights	earth	romantic	story	love	friendship
D1	1	1	1	1	1	0	0	0	0	0	0	0
D2	0	1	0	0	1	1	1	1	0	0	0	0
D3	0	0	0	0	0	0	0	0	1	1	1	1

Step 4: Similarity Measurement (Cosine Similarity)

$$\text{Cosine Similarity}(A,B) = \frac{A \cdot B}{\| A \| \| B \|}$$

$$\mathrm{D}_1.\mathrm{D}_2 = 2, ||D_1|| = \sqrt{5}, ||D_2|| = \sqrt{5},$$

$$\text{Cosine Similarity}(D_1, D_2) = \frac{2}{\sqrt{5}.\sqrt{5}} = \frac{2}{5} = 0.4$$

Similarity Scores
Similarity (D1, D2) ≈ 0.4 (High similarity), Similarity (D1, D3) = 0 and Similarity (D2, D3) = 0
Step 5: Recommendation Logic

- D1 and D2 share common themes: **hero + aliens + saving Earth**
- D3 is **romance-based**, unrelated to the first two

Practical-1: News Article Similarity Analysis Using TF-IDF
Step 1: Converting Text into Vectors

Method: TF–IDF (Frequency-Based Representation)

1. Words are treated as dimensions
2. Each document becomes a vector:

$$\mathbf{d} = (w_1, w_2, \dots, w_n)$$

TF (Term Frequency):

$$\mathrm{TF}(t, d) = \frac{\text{Number of times term } t \text{ appears in } d}{\text{Total terms in } d}$$

IDF (Inverse Document Frequency): $\mathrm{IDF}(t) = \log\left(\frac{N}{\mathrm{DF}(t)}\right)$

TF-IDF: $\text{TF-IDF}(t, d) = \mathrm{TF} \times \mathrm{IDF}$

Interpretation (Very Important for Exams & AI Context)

- The similarity is **zero** because the two documents **share no common weighted terms** after TF–IDF.
- Even though both documents contain **"AI"**, its **TF–IDF weight is zero**, since it appears in **all documents**.
- Hence, TF–IDF correctly identifies that the documents are **conceptually related but lexically different**.

Step 2: Measuring Similarity

Once documents are vectors:

$$\text{Similarity} = \frac{\mathbf{d}_1 \cdot \mathbf{d}_2}{\parallel \mathbf{d}_1 \parallel \ \parallel \mathbf{d}_2 \parallel}$$

Example:1

Document 1: "Mathematics is vital for AI"
Document 2: "AI depends on mathematical models"

Given Documents

- **Document 1 (D_1):** *"Mathematics is vital for AI"*
- **Document 2 (D_2):** *"AI depends on mathematical models"*

Step 1: Text Preprocessing

(a) Convert to lowercase

(b) Remove punctuation

(c) Tokenization

(d) Remove stop words (like *is, for, on*)

Processed Documents

D_1: mathematics, vital, ai

D_2: ai, depends, mathematical, models

Step 2: Vocabulary Construction

List **all unique terms** from both documents.

$$V = \{\text{mathematics,vital,ai,depends,mathematical,models}\}$$

Total documents: $N = 2$

Step 3: Term Frequency (TF)

$$\text{TF}(t, d) = \frac{\text{Number of times term } t \text{ appears in } d}{\text{Total terms in } d}$$

TF for Document 1

Total words in D_1 = 3

Term	mathematics	vital	ai	depends	mathematical	models
Frequency	1	1	1	0	0	0
TF	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	0	0

TF for Document 2

Total words in D_2 = 4

Term	ai	depends	mathematical	models	mathematics	vital
Frequency	1	1	1	1	0	0
TF	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	0	0

Step 4: Document Frequency (DF)

$$\text{DF}(t) = \text{Number of documents containing term } t$$

Term	mathematics	vital	ai	depends	mathematical	models
DF	1	1	2	1	1	1

Step 5: Inverse Document Frequency (IDF)

$$\text{IDF}(t) = \log\left(\frac{N}{\text{DF}(t)}\right)$$

Using **natural logarithm (ln)**:

IDF Values

Term	IDF
mathematics	**(log($\frac{2}{1}$) = 0.693)**
vital	**(log($\frac{2}{1}$) = 0.693)**
ai	**(log($\frac{2}{2}$) = 0)**
depends	**(log($\frac{2}{1}$)= 0.693)**
mathematical	**(log($\frac{2}{1}$) = 0.693)**
models	**((log($\frac{2}{1}$) = 0.693)**

Step 6: TF–IDF Calculation

$$\text{TF–IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

TF–IDF for Document 1:

Term	TF	IDF	TF–IDF
mathematics	$\frac{1}{3}$	**0.693**	**0.231**
vital	$\frac{1}{3}$	**0.693**	**0.231**
ai	$\frac{1}{3}$	**0**	**0**
depends	**0**	**0.693**	**0**
mathematical	**0**	**0.693**	**0**
models	**0**	**0.693**	**0**

TF–IDF for Document 2

Term	TF	IDF	TF–IDF
ai	$\frac{1}{4}$	**0**	**0**
depends	$\frac{1}{4}$	**0.693**	**0.173**

Term	TF	IDF	TF–IDF
mathematical	$\frac{1}{4}$	**0.693**	**0.173**
models	$\frac{1}{4}$	**0.693**	**0.173**
mathematics	**0**	**0.693**	**0**
vital	**0**	**0.693**	**0**

Step 7: Final TF–IDF Vectors

Document 1 Vector

$$D_1 = (0.231,\ 0.231,\ 0,\ 0,\ 0,\ 0)$$

Document 2 Vector

$$D_2 = (0,\ 0,\ 0,\ 0.173,\ 0.173,\ 0.173)$$

Cosine Similarity Formula $\cos(\theta) = \frac{D1 \cdot D2}{||D1||.||D2||}$

As $D1.D2 = 0$.

Therefore, $\cos(\theta) = 0$

Conclusion:

What Cosine Similarity Actually Measures

Cosine similarity measures the **angle between two vectors**, **not their length**.

$$\text{Cosine Similarity} = \cos(\theta)$$

- θ= angle between document vectors
- It checks **directional similarity**, not size

Geometric Interpretation (Very Important)

Think of each document as a **point/vector in high-dimensional space**.

Case 1: Cosine similarity = 1

$$\theta = 0^\circ$$

- Vectors point in the **same direction**
- Documents are **identical in content distribution**
- Same important words with similar weights

Meaning: Documents are *almost the same*

Case 2: Cosine similarity between 0 and 1

$$0^\circ < \theta < 90^\circ$$

- Some overlap in important terms
- Partial similarity

Meaning: Documents are *related* to some extent

Case 3: Cosine similarity = 0

$$\theta = 90^\circ$$

- Vectors are **orthogonal**
- No overlapping **important features**

Meaning:
Documents are *not similar in meaningful terms*

Very Important Clarification

Turnitin / iThenticate similarity % ≠ Cosine similarity

They **do NOT use TF–IDF cosine similarity** the way we compute in NLP textbooks.

Cosine similarity → **vector-based mathematical similarity**
Turnitin/iThenticate → **text overlap / matching similarity**

Then what is the purpose of cosine similarity?

Main Uses of Cosine Similarity (Very Important)

Information Retrieval (Search Engines)

When you search:
"AI mathematical models"
The search engine:

- Converts query + documents into vectors
- Computes cosine similarity
- Ranks documents by **highest cosine similarity**

Without cosine similarity, Google would not work.