# Vector Databases

- Traditional Databases are not always good for finding "Similar" Entities

- Resolution: Vector Database

## Vector Representations: (Embedding)

An embedding of an Entity is a high-Dimensional vector of Values (Real values)

$$t_i \longrightarrow \vec{v}_i \in \mathbb{R}^d$$

such that given a distance function $\Delta(t_i, t_j)$

$$\text{dist}(\vec{v}_i, \vec{v}_j) \sim \Delta(t_i, t_j)$$

$\cos\measuredangle(\vec{v}_i, \vec{v}_j)$ : Similari- function

Vector Database:

A Collection of Tables that every table T is in form $\{\{A_1, \dots, A_d\}, \{v_1, \dots v_d\}\}$

Traditional attributes

vector Representation

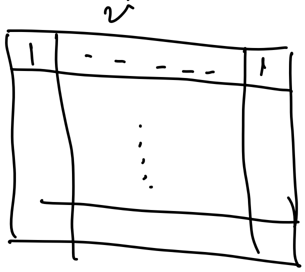
Example:

Animal

| ID | name | Category | Embedding |
| --- | --- | --- | --- |
| 1 | Dog | 1 | 1...0..1 |
| 2 | Cat | 2 | 0...01 |
| | ⋮ | | |
| n | Lion | 2 | 11...1001 |

Simplified Vector DB

$\vec{v}$



Every row is an Embedding.

Challenges (Compared to DB)

1 – Vague Similarity:

How to Compare two objects?

2 – Expensive Comparisons

$O(d)$

3 – High Dimension

$d \sim (100 - 10{,}000)$

4 – Lack of Structure:

– The Columns do not have any meanings independently.

⇒ Indexing Strategies based on Ordering / Hashing are not effective.

5 – Different Goals, Target Users

– Traditional queries are initiated & Understood by Human beings

– VectorDB 〃 〃 〃 〃 〃 by machine

How to Combine these Perspectives?

## Traditional DBs:

Query Types

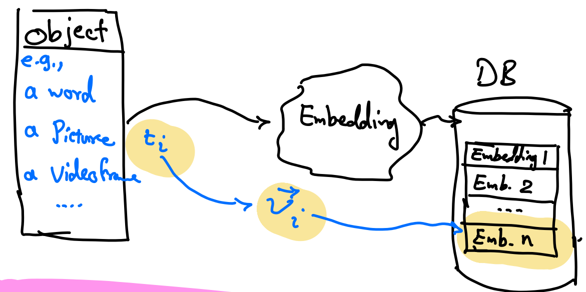
- DDL: Create Table / Create Shema
- DML: Insert / Del / Update; Select ....

## Vector DB

Query Types

- Data Man. : Insert / delete
- Similarity Search
  - objects similar to a given query object
  - $f(D, q)$ ↳ query Point
- Constrained (Predicated) queries
  - Find similar objects from the ones that satisfy a condition
  - ↳ Extended VDBs
- Multi-Vector Queries
  - ↳ Single vector → Multiple vectors
  - M → M
  - M → S

Insert:




Who provides the details of the Embedding Transformation?

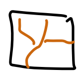
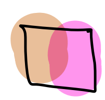
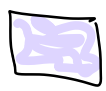
$$\vec{v}_i = f(t_i)$$

e.g.,

word Embedding: (word2vec)

| Sentence | Embedding ($f(t_i)$) | | | |
|---|---|---|---|---|
| we | 2.5 | 3 | 4.8 | 11 |
| love | 1 | 1 | 2.5 | 9.2 |
| UIC | 0.23 | 4.8 | 9.2 | 1.3 |

n×d table of Embeddings

| Image | Embedding ($f'(t_i)$) | | | | |
|---|---|---|---|---|---|
|  | 1 | 2 | ..... | 12 |
|  | 3.8 | 9 | ---- | 0.9 |
| ... | ... | | | |
|  | 8.5 | 10.3 | .... | 5.2 |

n×d′ table of embeddings

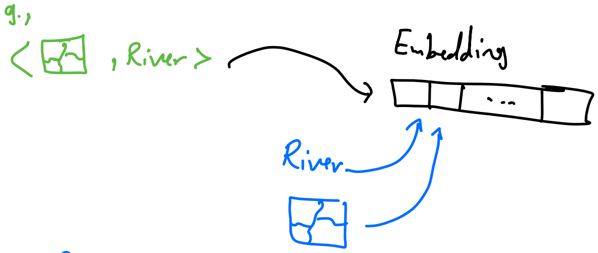
Embeddings in different "Embedding Spaces" are not comparable

## Jointly-Trained Embeddings

The Training Process is done on <object, Tag>

Tag: Textual

e.g.,

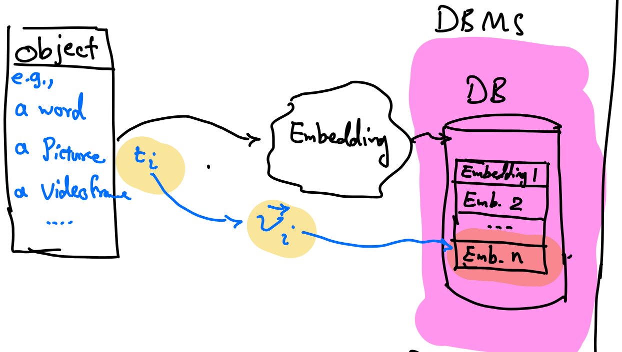



CLIP. Jointly Trained on Image & Textual Tags

---

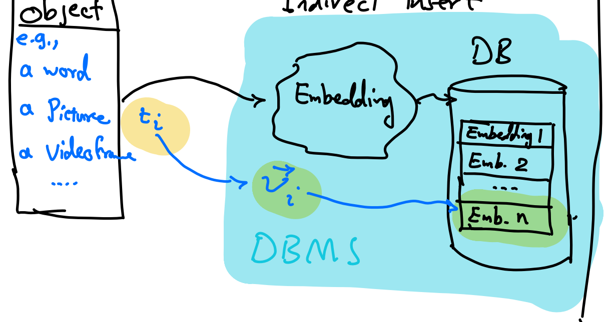
**Insert Types**

- Direct: The user is responsible for providing the Embedding
- Indirect: The DB is in charge of providing the Embedding




Direct Insert

Indirect Insert




## Direct Insert:

issues

1. Quality of the embedder left to the user.
2. The logical associations are unknown.
   → The DB cannot detect if the comparison is legit

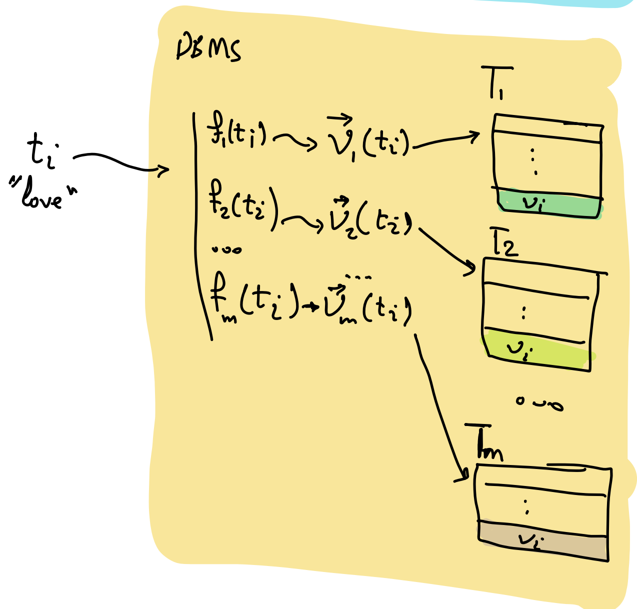
Benefits

1. more freedom to user
2. Easier to adapt for DBMS

---

## Indirect Insert:

Can use multiple Embedding



## Similarity Metrics:

* I will use the words distance & Similarity interchangably

e.g., $Sim(a,b) = 1 - dist(a,b)$

---

## Similarity Types

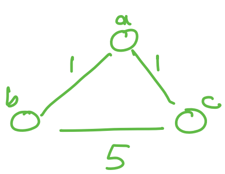
- Metric:
- Non-metric:

for a distance function $\Delta$ to be metric, it should satisfy the following properties

1- Identity: $\Delta(a,a) = 0$
2- Positivity: $\Delta(a,b) \geq 0$
3- Simmetry: $\Delta(a,b) = \Delta(b,a)$
4- Triangular Ineq:

$$\Delta(a,b) \leq \Delta(a,c) + \Delta(b,c)$$

e.g.

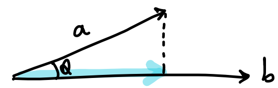



does not satisfy the Triangular Ineq

$w(b,c) = 5 > 1+1$

## Distance/Similarity Measures

① Inner Product (Dot Product)




a $[1, 1, 3]$
b $[2, 0, 1]$

$$a \cdot b = \langle a, b \rangle = \sum a_i b_i$$

$$a \cdot b = 2\times1 + 1\times0 + 3\times1 = 5$$

$$a \cdot b = \|a\| \cdot \|b\| \cos\theta$$

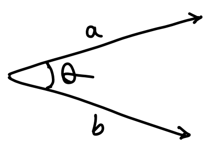
the projection of b on a

Not in metric

- Identity ✗
- Positivity ✗
- Simmetry ✓
- Triangular Ineq ✗

---

## Cosine Similarity




$Cosine(a,b) = Cos(\theta)$

$$Cos(a,b) = \frac{a}{\|a\|} \cdot \frac{b}{\|b\|}$$

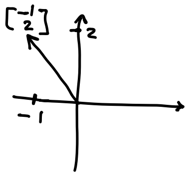
$\|a\|_2 = \|a\| = \sqrt{a \cdot a}$

$\|a\|_p = (\underbrace{a \cdot a \cdots a}_{p \text{ times}})^{1/p}$

e.g.,

$a = [-1, 0, 3, 5]$

$b = [2, 1, -1, -3]$




$\cos(a, b) = ?$

$\|a\| = \sqrt{a \cdot a} = \sqrt{1+0+9+25} = \sqrt{35}$

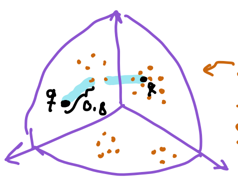
$\|b\| = \sqrt{b \cdot b} = \sqrt{4+1+1+9} = \sqrt{15}$

$\frac{a}{\|a\|} = \frac{[-1, 0, 3, 5]}{\sqrt{35}}, \quad \frac{b}{\|b\|} = \frac{[2, 1, -1, -3]}{\sqrt{15}}$

$\Rightarrow \text{Cosine}(a, b) = \frac{[-1, 0, 3, 5]}{\sqrt{35}} \cdot \frac{[2, 1, -1, -3]}{\sqrt{15}}$

$= \frac{(-2 + 0 + (-3) + (-15))}{\sqrt{525}}$

$= -20/\sqrt{525} \approx -0.8$




← Points are on the Surface of the Sphere

Cosine is <u>Not</u> Metric.

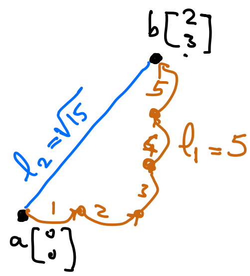
$\ell_p$ - norm measures

$\|a - b\|_p = \left(\sum |a_i - b_i|\right)^{1/p}$

- $\ell_1$ - norm
- $\ell_2$ - norm : Euclidean Distance
- $\ell_0$ - norm
- $\ell_\infty$ - norm

$\ell_1$ - norm : $\sum_{i=1}^{d} |a_i - b_i|$

$\ell_2$ - norm : $\sqrt{\sum (a_i - b_i)^2}$




$\ell_0$ - norm $= \left(\sum |a_i - b_i|^0\right)^{\cancel{\infty}\ 1}$

= The number of non-zero Values

$\ell_\infty$ - norm : The <u>maximum</u> absolute value of $|a_i - b_i|$

$= \max |a_i - b_i|$

e.g.,

$a = [-1, 0, 3, 5]$

$b = [2, 1, -1, -3]$

$$\ell_1 = \underset{3}{\cancel{(2+1)}} + \underset{1}{\cancel{(1-0)}} + \underset{4}{\cancel{(3+1)}} + \underset{8}{\cancel{(5-(-3))}} = 16$$

$$\ell_2 = \sqrt{\underset{9}{\cancel{(2+1)^2}} + \underset{1}{\cancel{1^2}} + \underset{16}{\cancel{4^2}} + \underset{64}{\cancel{8^2}}}$$

$$= \sqrt{90} \qquad = 3\sqrt{10}$$

$$\ell_0 = \mathbb{1}(3, 1, 4, 8) = 4$$

$$\ell_\infty = \max(3, 1, 4, 8) = 8$$

Lp-norm is in Metric.