# Agenda

- Vector space model
- Cosine similarity
- Jaccard index
- Edit distance

# The Vector Space Model

- Assume $t$ distinct terms remain after preprocessing; call them index terms or the vocabulary.
- These "orthogonal" terms form a vector space.

  $$\text{Dimension} = t = |\text{vocabulary}|$$

- Each term, $i$, in a document or query, $j$, is given a real-valued weight, $w_{ij}$.
- Both documents and queries are expressed as t-dimensional vectors:

  $$d_j = (w_{1j}, w_{2j}, \ldots, w_{tj})$$

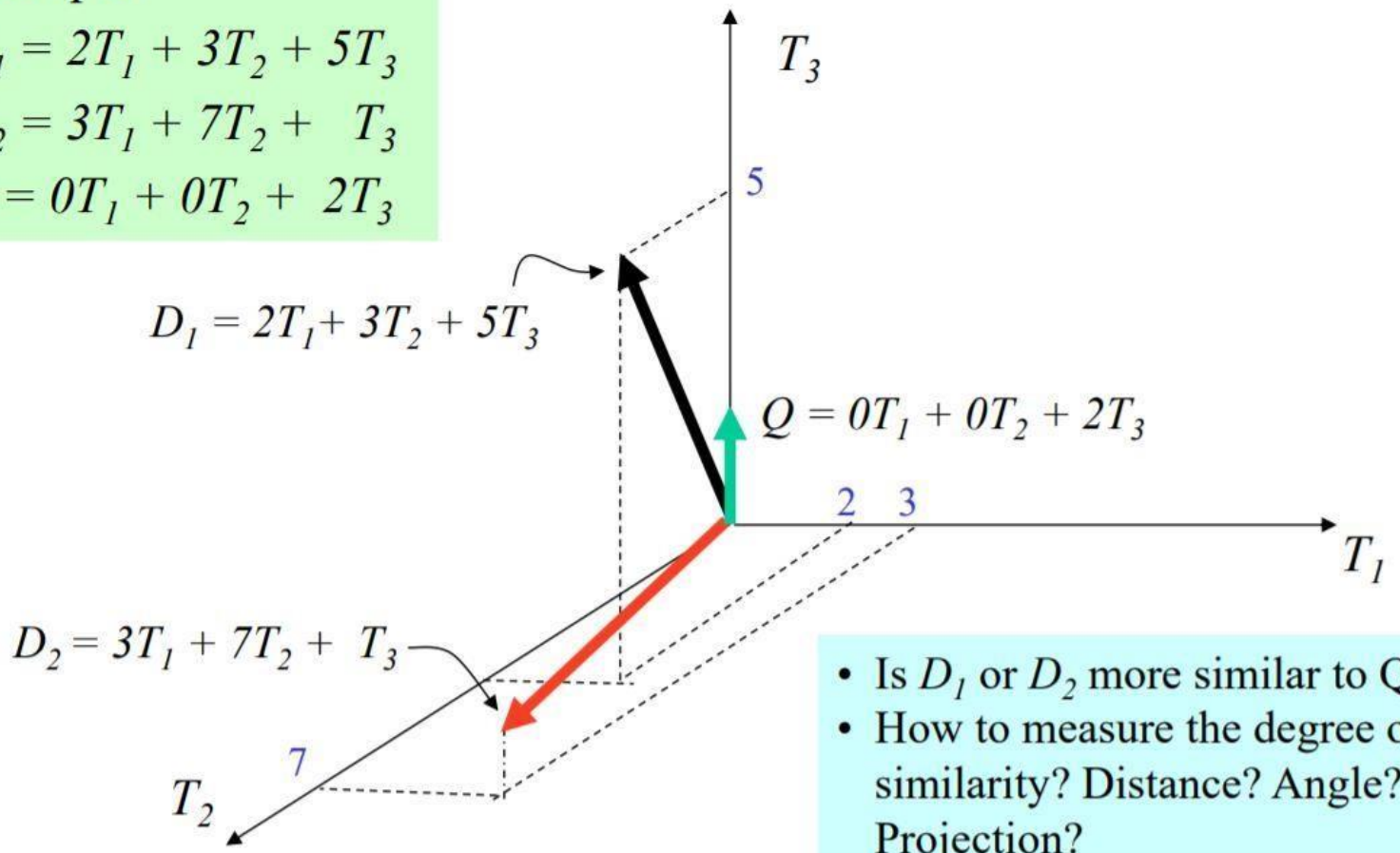
# Graphical representation

Example:

$D_1 = 2T_1 + 3T_2 + 5T_3$

$D_2 = 3T_1 + 7T_2 + T_3$

$Q = 0T_1 + 0T_2 + 2T_3$




- Is $D_1$ or $D_2$ more similar to Q?
- How to measure the degree of similarity? Distance? Angle? Projection?

# Document collection

- A collection of *n* documents can be represented in the vector space model by a term-document matrix.
- An entry in the matrix corresponds to the "weight" of a term in the document; zero means the term has no significance in the document or it simply doesn't exist in the document.

$$\begin{array}{c|cccc} & T_1 & T_2 & \ldots & T_t \\ D_1 & w_{11} & w_{21} & \ldots & w_{t1} \\ D_2 & w_{12} & w_{22} & \ldots & w_{t2} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ D_n & w_{1n} & w_{2n} & \ldots & w_{tn} \end{array}$$

# Term Weights: Term Frequency (Recap)

- A typical combined term importance indicator is *tf-idf weighting*:

$$w_{ij} = tf_{ij}\, idf_i = tf_{ij} \log_2 (N/ df_i)$$

- A term occurring frequently in the document but rarely in the rest of the collection is given high weight.
- Many other ways of determining term weights have been proposed.
- Experimentally, *tf-idf* has been found to work well.

# Example (TF-IDF)

Given a document containing terms with given frequencies:

A(3), B(2), C(1)

Assume collection contains 10,000 documents and

document frequencies of these terms are:

A(50), B(1300), C(250)

Then:

A: tf = 3/3; idf = log(10000/50) = 5.3; tf-idf = 5.3

B: tf = 2/3; idf = log(10000/1300) = 2.0; tf-idf = 1.3

C: tf = 1/3; idf = log(10000/250) = 3.7; tf-idf = 1.2

# Similarity Measure - Inner Product

- Similarity between vectors for the document $\boldsymbol{d}_i$ and query $\boldsymbol{q}$ can be computed as the vector inner product:

$$\text{sim}(\boldsymbol{d}_j, \boldsymbol{q}) = \boldsymbol{d}_j \bullet \boldsymbol{q} = \sum_{i=1}^{t} w_{ij} \cdot w_{iq}$$

where $w_{ij}$ is the weight of term $i$ in document $j$ and $w_{iq}$ is the weight of term $i$ in the query

- For binary vectors, the inner product is the number of matched query terms in the document (size of intersection).
- For weighted term vectors, it is the sum of the products of the weights of the matched terms.

# Inner Product - Examples

Binary:

| | retrieval | database | architecture | computer | text | management | information |
|---|---|---|---|---|---|---|---|
| D | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| Q | 1 | 0 | 1 | 0 | 0 | 1 | 1 |

Size of vector = size of vocabulary = 7
0 means corresponding term not found in document or query

$sim(D, Q) = 3$

Weighted:

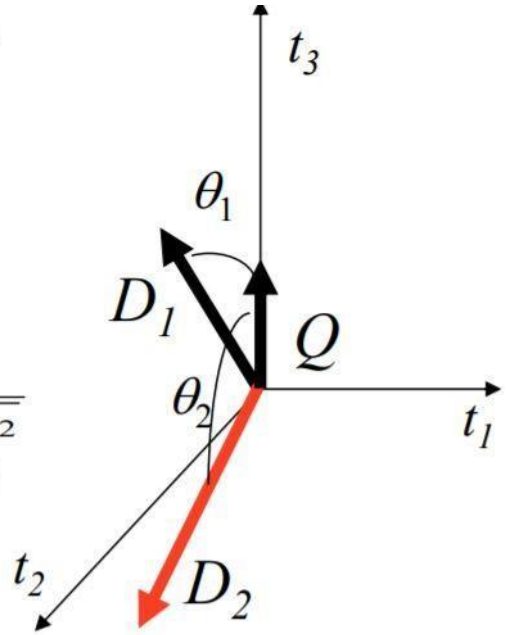
$D_1 = 2T_1 + 3T_2 + 5T_3 \qquad D_2 = 3T_1 + 7T_2 + 1T_3$
$Q = 0T_1 + 0T_2 + 2T_3$

$sim(D_1, Q) = 2*0 + 3*0 + 5*2 = 10$
$sim(D_2, Q) = 3*0 + 7*0 + 1*2 = 2$

# Cosine SimilarityMeasure

- Cosine similarity measures the cosine of the angle between two vectors.
- Inner product normalized by the vector lengths.

$$\text{CosSim}(\boldsymbol{d}_j, \boldsymbol{q}) = \frac{\vec{d}_j \cdot \vec{q}}{\left|\vec{d}_j\right| \cdot \left|\vec{q}\right|} = \frac{\sum_{i=1}^{t} (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^{t} w_{ij}^{\,2} \cdot \sum_{i=1}^{t} w_{iq}^{\,2}}}$$




$D_1 = 2T_1 + 3T_2 + 5T_3$ $\quad \text{CosSim}(D_1, Q) = 10 / \sqrt{(4+9+25)(0+0+4)} = 0.81$

$D_2 = 3T_1 + 7T_2 + 1T_3$ $\quad \text{CosSim}(D_2, Q) = 2 / \sqrt{(9+49+1)(0+0+4)} = 0.13$

$Q = 0T_1 + 0T_2 + 2T_3$

$D_1$ is 6 times better than $D_2$ using cosine similarity but only 5 times better using inner product.

# Exercise

- A user is browsing 4 websites, he wants to find the similarity between each website content. After pre-processing is finished, the content of the 4 websites is as follows:

  - $\mathbf{W_1}$ = information media science
  - $\mathbf{W_2}$ = media science lebanese
  - $\mathbf{W_3}$ = media lebanese science information science
  - $\mathbf{W_4}$ = lebanese media lebanese science

- Write the binary and term-weighted matrix
- Calculate the similarity of $W_1$ with $W_2$ and $W_3$
- Calculate the similarity of $W_1$ with $W_4$
- What is the most similar document to $W_1$
- Draw the vectors in a 2 dimentional (2D) space, with the correct angle between each vector.

# Solution

- **Binary matrix**

| | Media | Science | Lebanese | Information |
|---|---|---|---|---|
| $W_1$ | 1 | 1 | 0 | 1 |
| $W_2$ | 1 | 1 | 1 | 0 |
| $W_3$ | 1 | 1 | 1 | 1 |
| $W_4$ | 1 | 1 | 1 | 0 |

- **Term weighted matrix**

| | Media | Science | Lebanese | Information |
|---|---|---|---|---|
| $W_1$ | 1 | 1 | 0 | 1 |
| $W_2$ | 1 | 1 | 1 | 0 |
| $W_3$ | 1 | 2 | 1 | 1 |
| $W_4$ | 1 | 1 | 2 | 0 |

# Solution

» **Vectors**

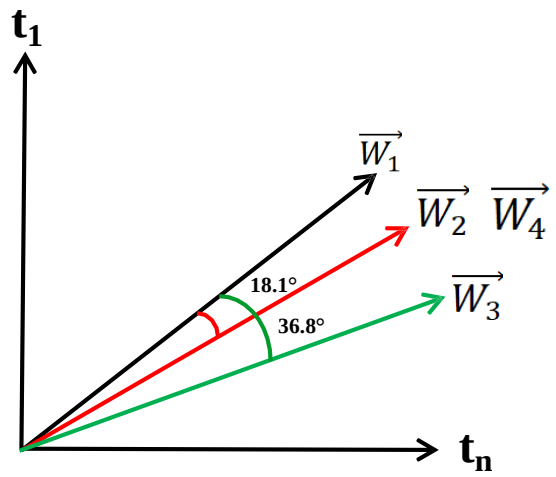
- $W_1 = 1T_1+1T_2+0T_3+1T_4$
- $W_2 = 1T_1+1T_2+0T_3+0T_4$
- $W_3 = 1T_1+2T_2+1T_3+1T_4$
- $W_4 = 1T_1+1T_2+2T_3+0T_4$

# Solution

- **Inner products**
  - $\text{sim}(W_1, W_2) = (1x1)+(1x1)+(0x1)+(1x0) = 2$
  - $\text{sim}(W_1, W_3) = 4$
  - $\text{sim}(W_1, W_4) = 2$
- Cosine similarity
  - $\cos(W_1, W_2) = \frac{4}{\sqrt{(1+1+1)(1+1+1)}} = \frac{2}{2.4} = 0.8$
  - $\cos(W_1, W_3) = 0.95$
  - $\cos(W_1, W_4) = 0.8$

# Solution

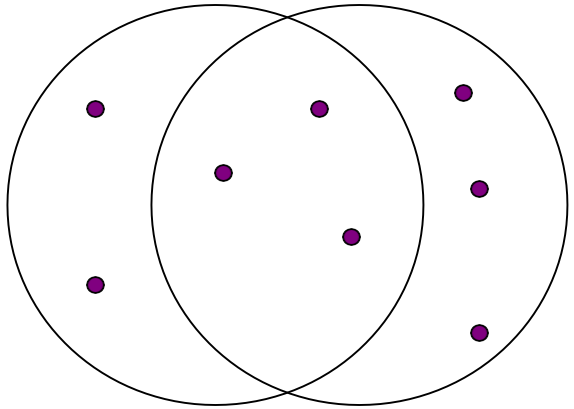
- **Vector space**
  - $\measuredangle(W_1, W_2) = \cos^{-1}(0.8) = 36.8$
  - $\measuredangle(W_1, W_3) = \cos^{-1}(0.95) = 18.1$
  - $\measuredangle(W_1, W_4) = \cos^{-1}(0.8) = 36.8$
- $W_2$ and $W_4$ are the most relevant documents to $W_1$.



# Jaccard Similarity

» The Jaccard similarity (Jaccard coefficient) of two sets $S_1$, $S_2$ is the size of their intersection divided by the size of their union.

- JSim $(C_1, C_2) = |C_1 \cap C_2| / |C_1 \cup C_2|$.



3 in intersection.
8 in union.
Jaccard similarity
= 3/8

- Extreme behavior:
  - Jsim(X,Y) = 1, iff X = Y
  - Jsim(X,Y) = 0 iff X,Y have no elements in common
- JSim is symmetric

# Jaccard Similarity between sets

- The distance for the documents


- JSim(D,D) = 3/5
- JSim(D,D) = JSim(D,D) =2/6
- JSim(D,D) = JSim(D,D) =3/9

# Edit Distance for strings

- The edit distance of two strings is the number of inserts and deletes of characters needed to turn one into the other.
- Example: x = abcde ; y = bcduve.
  - Turn *x* into *y* by deleting a, then inserting u and v after d.
  - Edit distance = 3.
- Minimum number of operations can be computed using dynamic programming
- Common distance measure for comparing DNA sequences