Jul – Nov 2022
IT458

Classical IR Models for Unstructured Text

Term Weight Modeling

Dr. Sowmya Kamath S, Dept of IT, NITK Surathkal

Term Weighting Models

- Introduced a novel concept of –
  - Evaluating how useful a term is, in a particular document.
  - E.g.
    - A term which appears in all documents of the corpus -- *completely useless for retrieval tasks.*
    - Term that occurs in very few documents – *important in determining doc. context.*

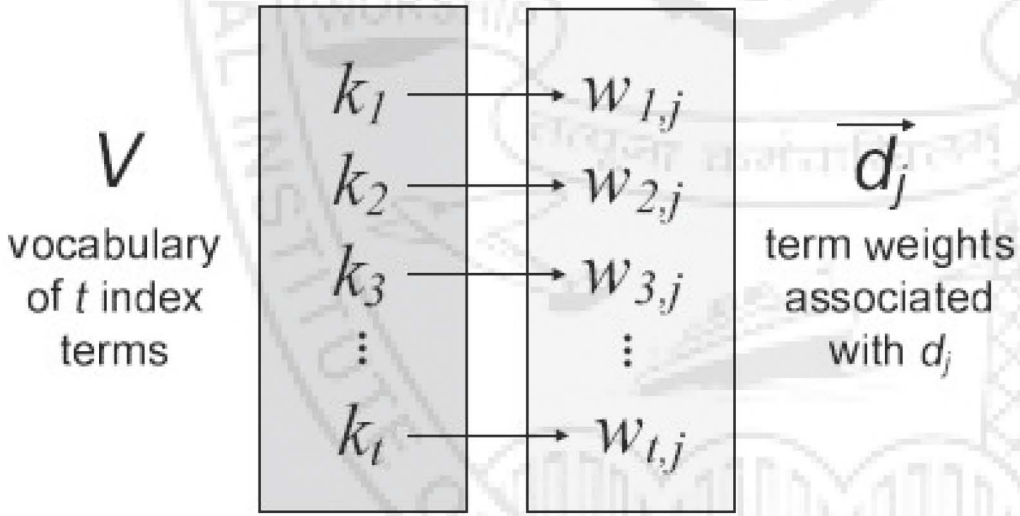
Term Weighting Models

- use some inherent properties of an index term for evaluating the importance of that term in a document.

- Term importance = **weight**
  - associated with each term in the vocabulary
  - Computed w.r.t each document in the corpus.

Term Weighting

- Let V = $\{k_1, k_2, \ldots, k_t\}$ where
  - k_i = index term;
  - d_j = document;
  - $w_{i,j}$ = weight associated with (k_i, d_j)



- Then, $d_j = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$ → **weighted vector** that gives the **weight** $w_{i,j}$ of **each term** $k_i \in V$ in the document d_j

Term Weighting

- Weight denoted as $\mathbf{w_{i,j}}$
  - quantifies the **importance** of the index term k_i in describing the **context** of document d_j.

$\mathbf{w_{i,j} > 0}$ for each term k_i that occurs in document d_j

$\mathbf{w_{i,j} = 0}$ if k_i does not appear in document d_j

Term Weighting – Notations used (summary)

- k_i - index term
- d_j - document

- V - **Vocabulary** represented as $\{k_1, k_2, \ldots, k_t\}$
- $w_{i,j}$ - **weight** associated with (k_i, d_j)

- N - number of documents in the corpus
- n_i - **Document Frequency** (No.of documents in which term k_i occurs)

- $f_{i,j}$ - **Frequency of term** k_i in document d_j
- F_i - **Frequency of occurrence of term** k_i in document corpus

Term Weighting – *Frequency of Term*

- **Frequency of term** $f_{i,j}$ in document d_j
  - Number of times term k_i occurs in document d_j

Term Weighting – *Frequency of Occurrence*

- **Frequency of occurrence F_i** of term k_i in collection is-

$$F_i = \sum_{j=1}^{N} f_{i,j}$$

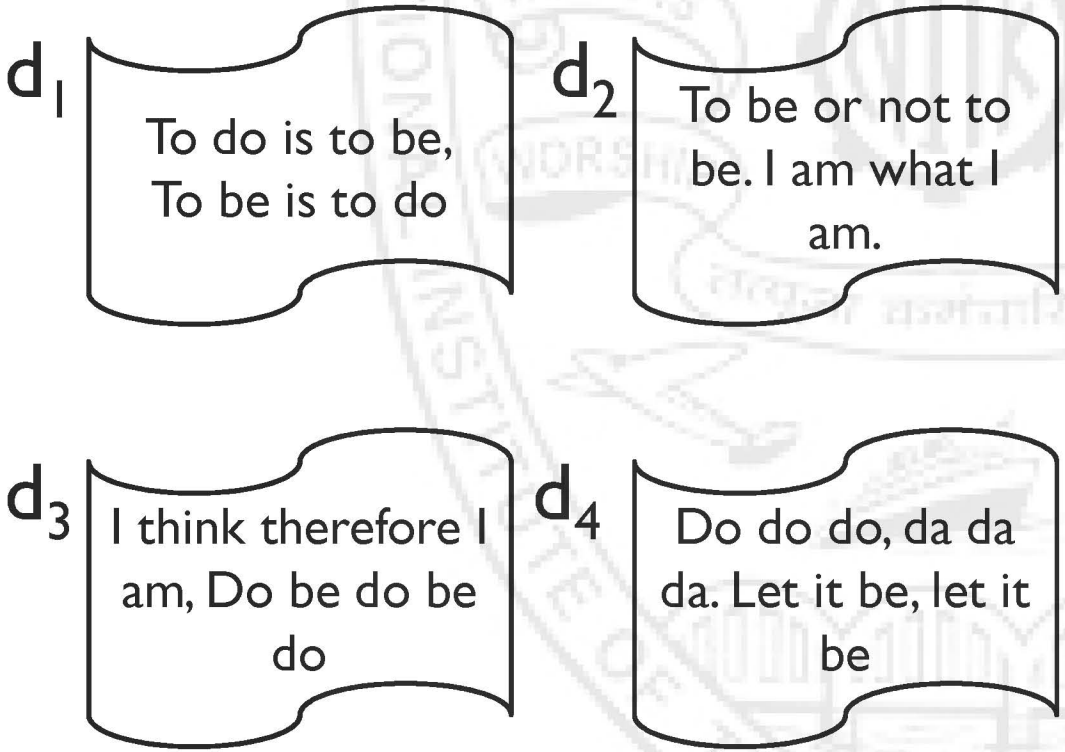
  - where N is the number of documents in the collection

Term Weighting – *Document Frequency*

- **Document frequency n_i** of a term k_i
  - the number of documents in which term k_i occurs
  - Note: $n_i \leq F_i$

Term Weighting – *Document Frequency*

- **Example**: Compute the values of the associated frequencies of a term k_i = **'do'.**



$f('do', d_1) = 2$
$f('do', d_2) = 0$
$f('do', d_3) = 3$
$f('do', d_4) = 3$
$F('do') = 8$
$n('do') = 3$

Weighting Schemes

- TF-IDF
  - foundation of the most popular term weighting schemes in IR.
- TF-IDF term weighting scheme –
  - Term frequency (TF)
  - Inverse document frequency (IDF)

TF-IDF weights

- TF-IDF term weighting scheme –
  - Term frequency (TF)

Log base is 2!!!!!!!

- **Term Frequency** $tf_{i,j} = \begin{cases} 1 + \log f_{i,j} & \texttt{if } f_{i,j} > 0 \\ 0 & \texttt{otherwise} \end{cases}$

where, the log is taken in base 2

TF-IDF weights

- Term Frequency Example

$$tf_{i,j} = \begin{cases} 1 + \log f_{i,j} & \texttt{if } f_{i,j} > 0 \\ 0 & \texttt{otherwise} \end{cases}$$

To do is to be.
To be is to do.

d_1

To be or not to be.
I am what I am.

d_2

I think therefore I am.
Do be do be do.

d_3

Do do do, da da da.
Let it be, let it be.

d_4

Vocabulary	
1	to
2	do
3	is
4	be
5	or
6	not
7	I
8	am
9	what
10	think
11	therefore
12	da
13	let
14	it

TF-IDF weights

- Term Frequency Example

$$tf_{i,j} = \begin{cases} 1 + \log f_{i,j} & \texttt{if } f_{i,j} > 0 \\ 0 & \texttt{otherwise} \end{cases}$$

To do is to be.
To be is to do.

d_1

To be or not to be.
I am what I am.

d_2

I think therefore I am.
Do be do be do.

d_3

Do do do, da da da.
Let it be, let it be.

d_4

Vocabulary		$tf_{i,1}$	$tf_{i,2}$	$tf_{i,3}$	$tf_{i,4}$
1	to	3	2	-	-
2	do	2	-	2.585	2.585
3	is	2	-	-	-
4	be	2	2	2	2
5	or	-	1	-	-
6	not	-	1	-	-
7	I	-	2	2	-
8	am	-	2	1	-
9	what	-	1	-	-
10	think	-	-	1	-
11	therefore	-	-	1	-
12	da	-	-	-	2.585
13	let	-	-	-	2
14	it	-	-	-	2

TF-IDF Weights

- **Inverse Document Frequency (IDF)**

$$idf_i = \log \frac{N}{n_i}$$

the log is taken in base 2

  - Where, N = size of the document collection,
    n_i = Document frequency of a term k_i (the number of documents in which it occurs)

* the IDF of a rare term is high, where as, that of a frequent term is likely to be low.

TF-IDF Weights

- IDF values of a sample collection

$$idf_i = \log \frac{N}{n_i}$$

To do is to be.
To be is to do.

d_1

To be or not to be.
I am what I am.

d_2

I think therefore I am.
Do be do be do.

d_3

Do do do, da da da.
Let it be, let it be.

d_4

	term	n_i	$idf_i = \log(N/n_i)$
1	to	2	1
2	do	3	0.415
3	is	1	2
4	be	4	0
5	or	1	2
6	not	1	2
7	I	2	1
8	am	2	1
9	what	1	2
10	think	1	2
11	therefore	1	2
12	da	1	2
13	let	1	2
14	it	1	2

TF-IDF Weighting Scheme

- Term weight = combination of IDF factors with term frequencies.
- Let $w_{i,j}$ be the term weight associated with the term k_i and the document d_j. Then, the TF-IDF weighting =

$$w_{i,j} = \begin{cases} (1 + \log f_{i,j}) \times \log \frac{N}{n_i} & \texttt{if } f_{i,j} > 0 \\ 0 & \texttt{otherwise} \end{cases}$$

TF-IDF Weighting Scheme

- Calculated TF-IDF - $w_{i,j} = \begin{cases} (1 + \log f_{i,j}) \times \log \frac{N}{n_i} & \texttt{if } f_{i,j} > 0 \\ 0 & \texttt{otherwise} \end{cases}$

To do is to be.
To be is to do.

d_1

To be or not to be.
I am what I am.

d_2

I think therefore I am.
Do be do be do.

d_3

Do do do, da da da.
Let it be, let it be.

d_4

	term	d_1	d_2	d_3	d_4
1	to	3	2	-	-
2	do	0.830	-	1.073	1.073
3	is	4	-	-	-
4	be	-	-	-	-
5	or	-	2	-	-
6	not	-	2	-	-
7	I	-	2	2	-
8	am	-	2	1	-
9	what	-	2	-	-
10	think	-	-	2	-
11	therefore	-	-	2	-
12	da	-	-	-	5.170
13	let	-	-	-	4
14	it	-	-	-	4

Variants of TF-IDF

- Five distinct variants for TF weights -

	tf weight
binary	{0,1}
raw frequency	$f_{i,j}$
log normalization	$1 + \log f_{i,j}$
double normalization 0.5	$0.5 + 0.5 \frac{f_{i,j}}{max_i f_{i,j}}$
double normalization K	$K + (1 - K) \frac{f_{i,j}}{max_i f_{i,j}}$

Variants of TF-IDF

- Five distinct variants for IDF weights -

	idf weight
unary	1
inverse frequency	$\log \frac{N}{n_i}$
inv frequency smooth	$\log(1+\frac{N}{n_i})$
inv frequeny max	$\log(1+\frac{max_i n_i}{n_i})$
probabilistic inv frequency	$\log \frac{N-n_i}{n_i}$

Some good questions to ponder ...

- What is the IDF of a term that occurs in every document? 0
- Can the TF-IDF weight of a term in a document exceed 1? yes
- Can the TF-IDF weight of a term be negative? no
- Why is the IDF of a term always finite? all variables finite

More reading...

- Jones, Karen Sparck. "Index term weighting." *Information storage and retrieval* 9.11 (1973): 619-633.

- Salton, G., Yang, C. S., & Yu, C. T. (1975). A theory of term importance in automatic text analysis. *Journal of the American society for Information Science*, *26*(1), 33-44.

Jul – Nov 2022
IT458

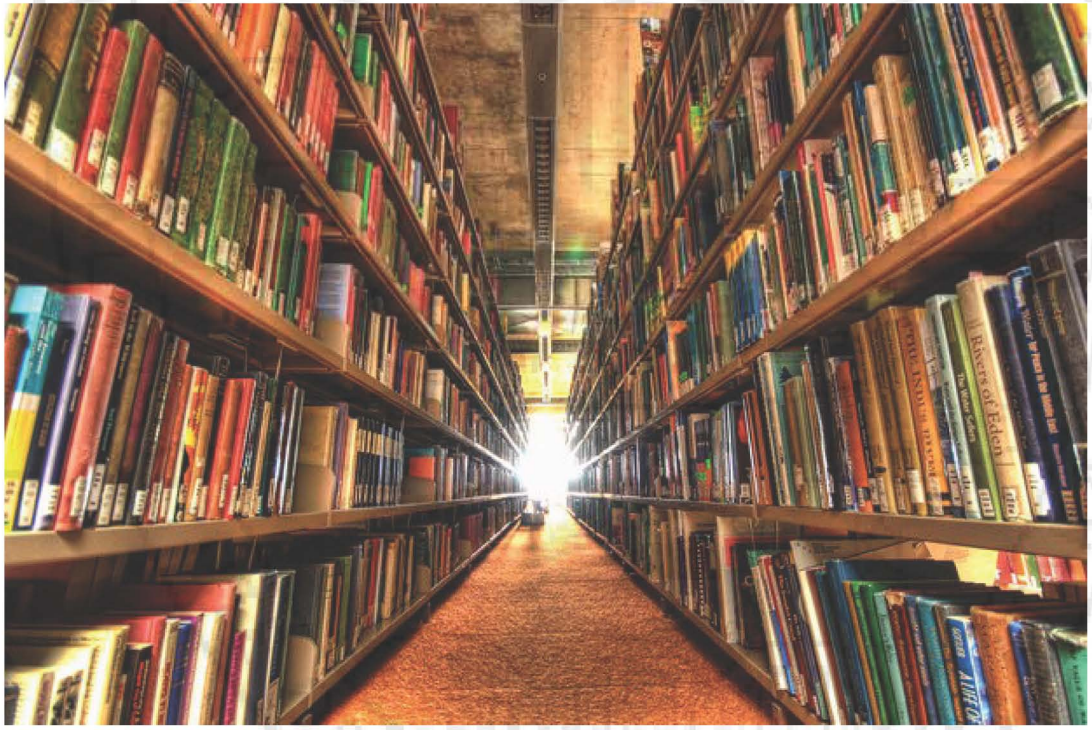
Classical IR Models for Unstructured Text

Vector Space Model

Dr. Sowmya Kamath S, Dept of IT, NITK Surathkal

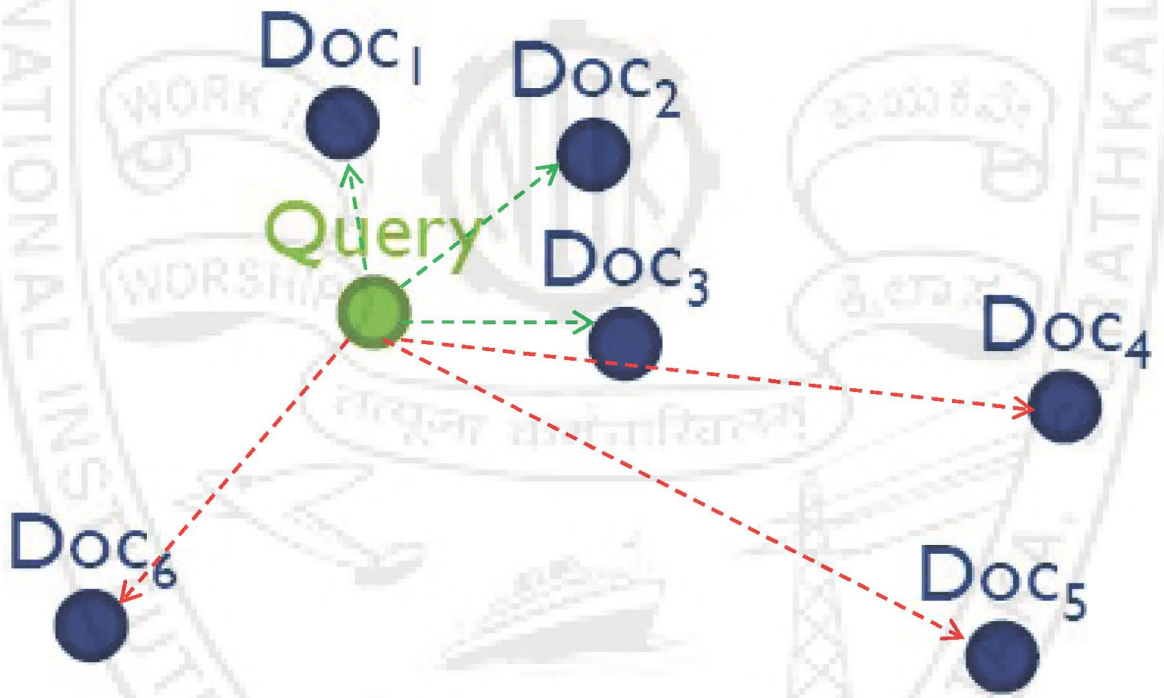
Information Spaces...

- Spatial structure of libraries:



- Idea: how to adapt this principle for information retrieval?

Information Spaces...



Solution:

- Represent documents and queries as points in an abstract semantic space.
- Measure similarity by proximity.

Vector Space Model (VSM)

- Documents and queries are represented as a point in *n*-dimensional real vector space $\mathbb{R}_n$
  - *n* is the size of the vocabulary.
  - Usually, *n* is very large.

- proposed by Prof. Gerard Salton, Cornell University (1975)

Ref: G Salton, A Wong, and C Yang. "A vector space model for automatic indexing." *Communications of the ACM* 18.11 (1975): 613-620.

Basic VSM – IR Model Formalisms

- Document d_j and query q → t-dimensional vectors
  - t - number of index terms in the vocabulary
- Query vector q → $(w_{1,q}, w_{2,q}, \ldots, w_{t,q})$
- Document vector d_j → $(d_{1,j}, d_{2,j}, \ldots, d_{t,j})$
  - $w_{t,q}$ and $d_{t,j}$ can assume positive values {0,1}
    - 1 if the term is present, 0 otherwise

Basic Vector Space Model

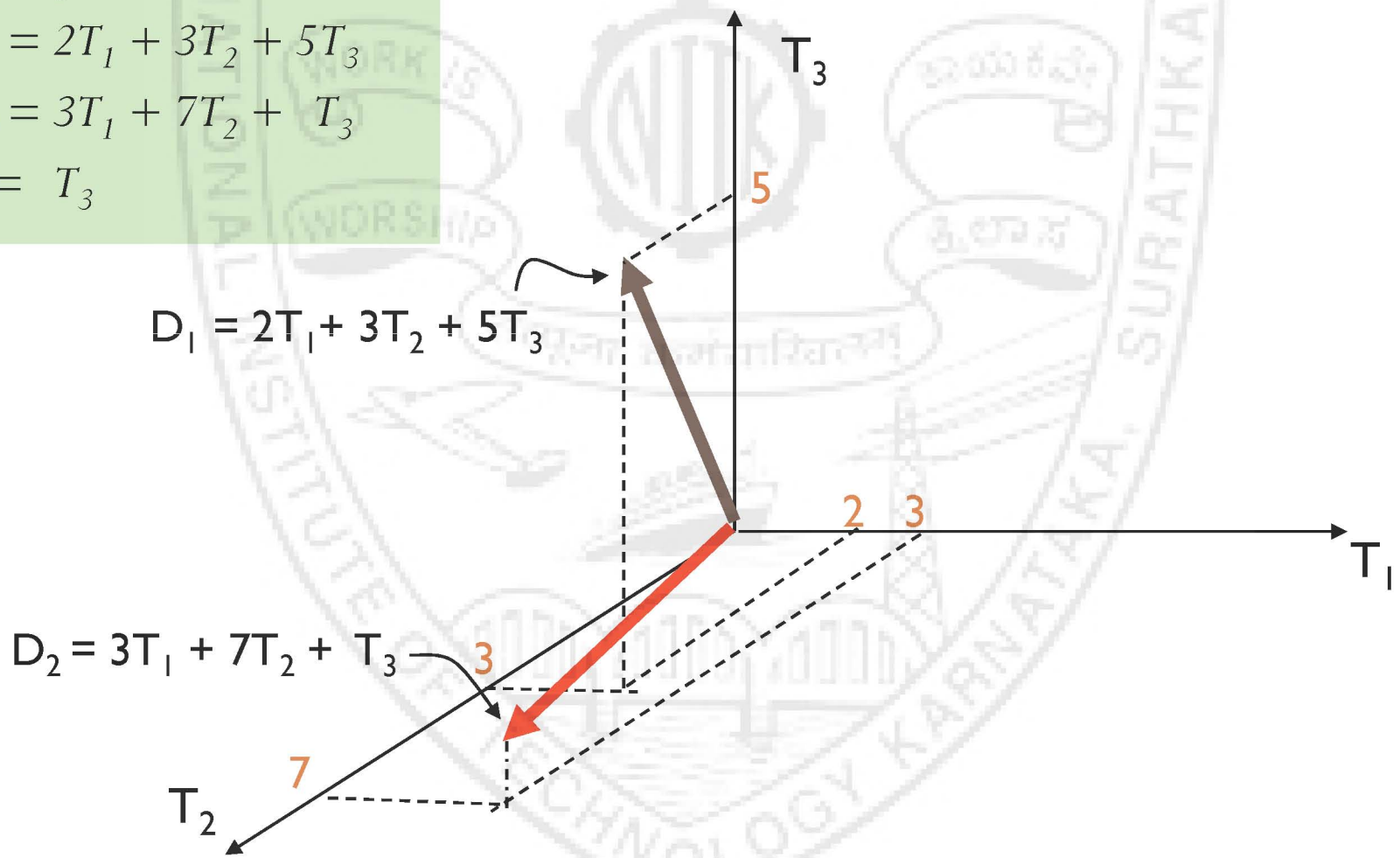
Example:

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

$D_2 = 3T_1 + 7T_2 + T_3$

$Q = T_3$

Graphical Representation



Basic Vector Space Model

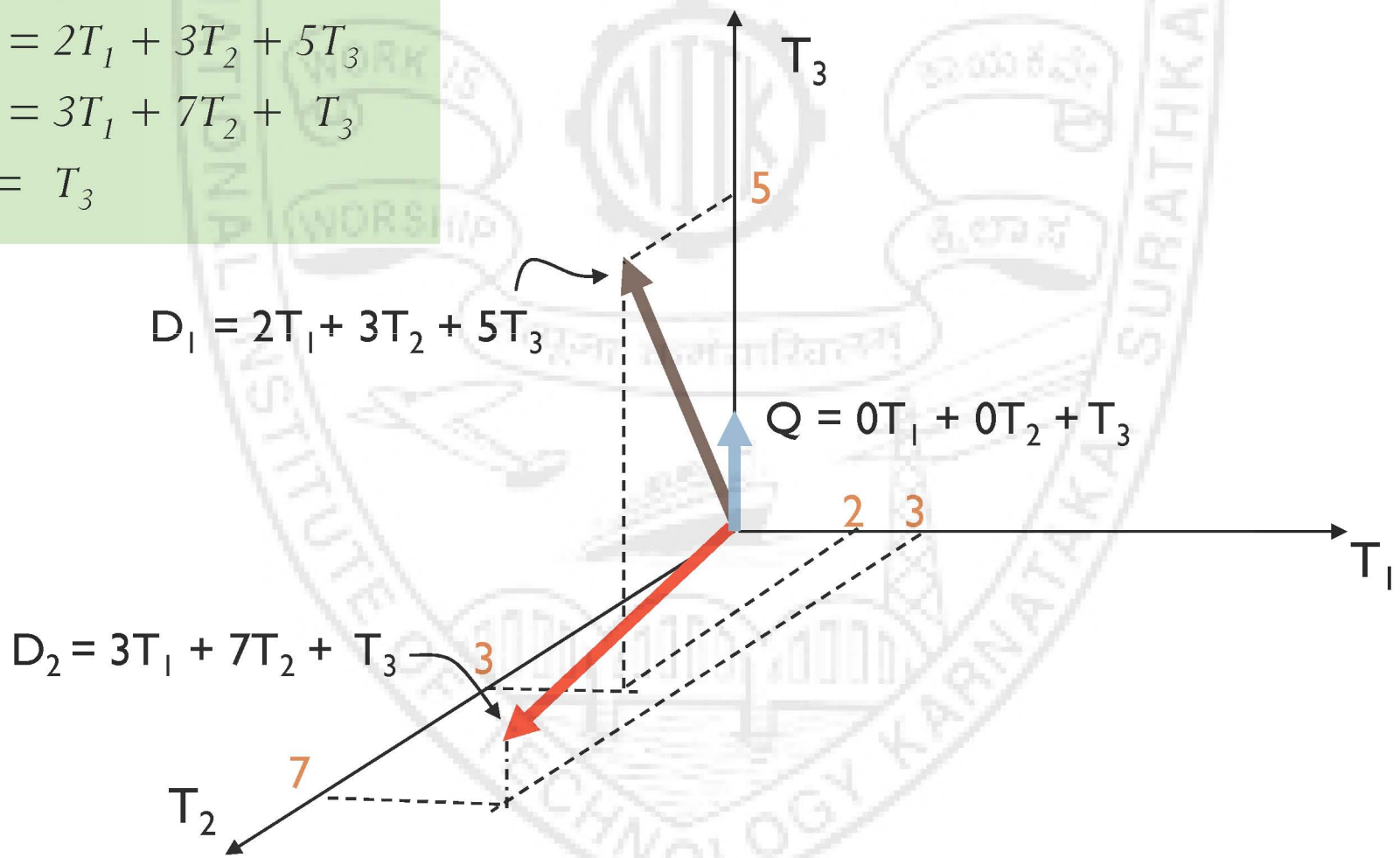
Example:

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

$D_2 = 3T_1 + 7T_2 + T_3$

$Q = T_3$

Graphical Representation



Basic Vector Space Model

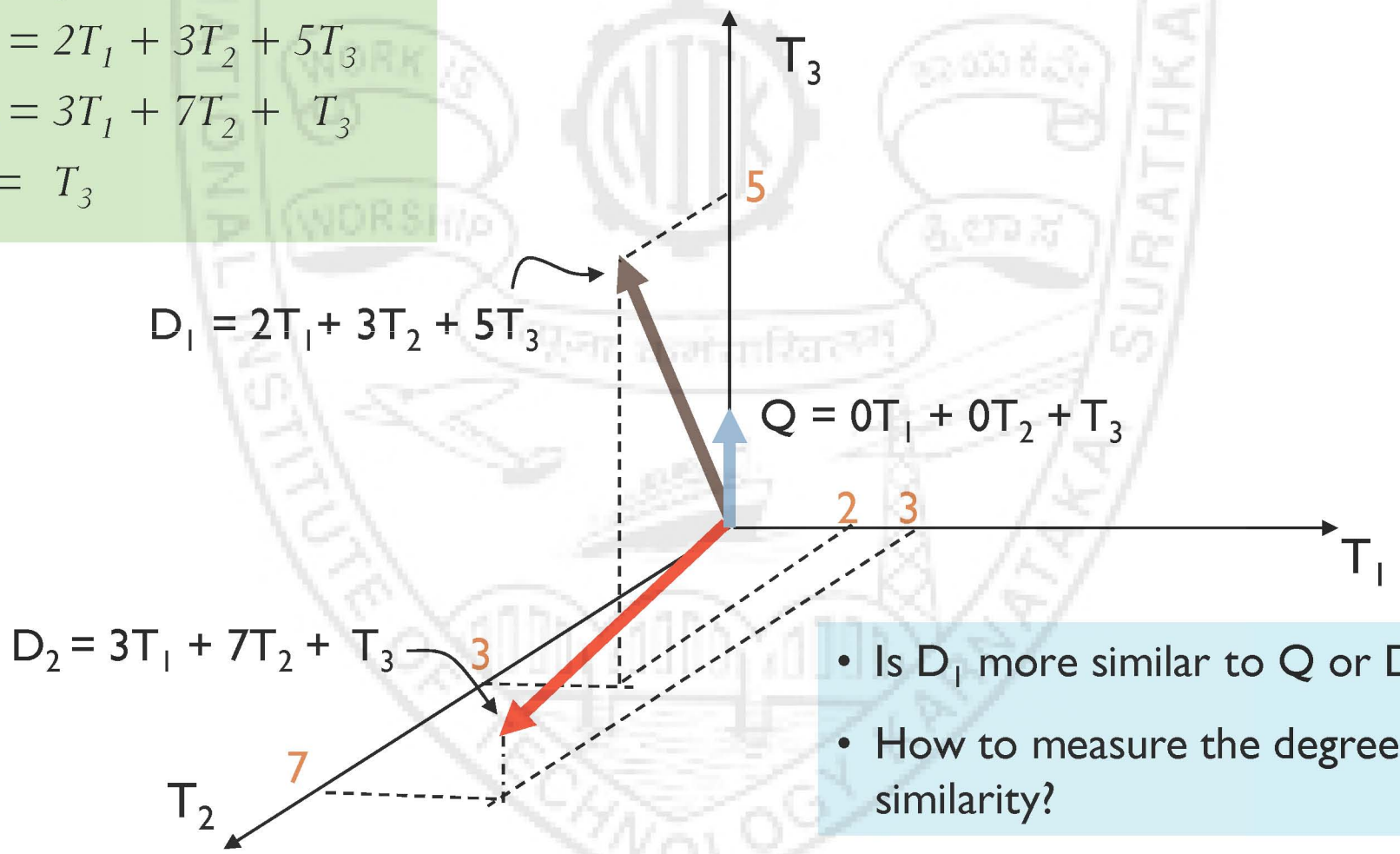
Example:

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

$D_2 = 3T_1 + 7T_2 + T_3$

$Q = T_3$

Graphical Representation



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

Basic Vector Space Model

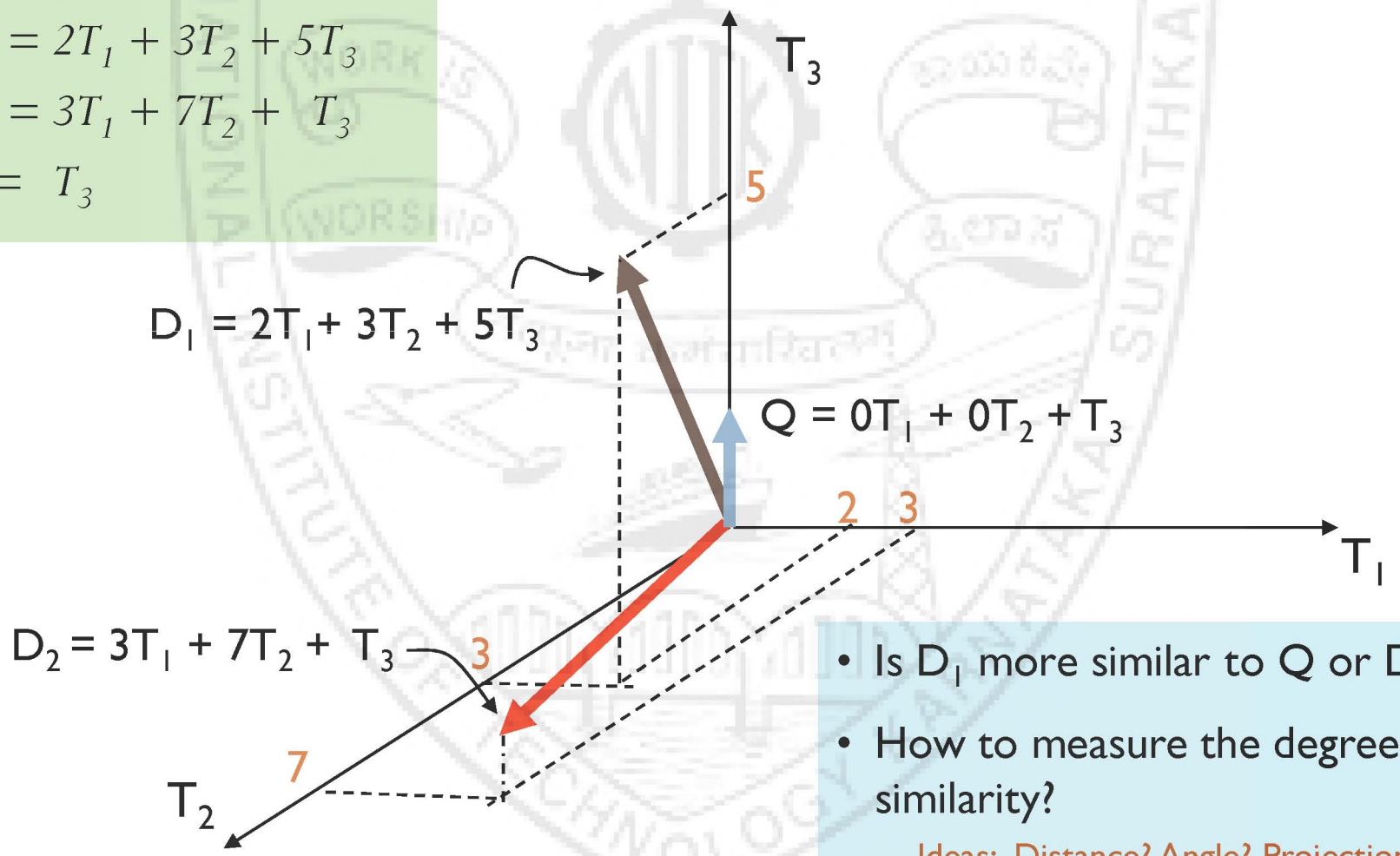
Example:

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

$D_2 = 3T_1 + 7T_2 + T_3$

$Q = T_3$

Graphical Representation



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

  Ideas: Distance? Angle? Projection?

VSM: Evaluating Vector Closeness

Question : How to evaluate vector closeness ??

- Idea: to use a measure of **distance** or **similarity** between documents

VSM: Evaluating Vector Closeness

- Approaches –
  - Using **"Distance"** for defining closeness through **proximity**
  - Using **"Similarity"** for defining closeness as **relevance**

VSM: Evaluating Vector Closeness

Using "Distance" for defining proximity

- A metric on a set X is a function $\mathbf{d : X \times X \rightarrow \mathbb{R}}$ with the properties -
  - $\mathbf{d(x, y) \geq 0}$, for any $\mathbf{x, y \in X}$ **(non-negativity)**
  - $\mathbf{d(x, y) = 0}$ **iff** $\mathbf{x = y}$, for any $\mathbf{x, y \in X}$ **(identity)**
  - $\mathbf{d(x, y) = d(y, x)}$, for any $\mathbf{x, y \in X}$ **(symmetry)**
  - $\mathbf{d(x, z) \leq d(x, y) + d(y, z)}$, for any $\mathbf{x, y, z \in X}$ **(triangle inequality)**

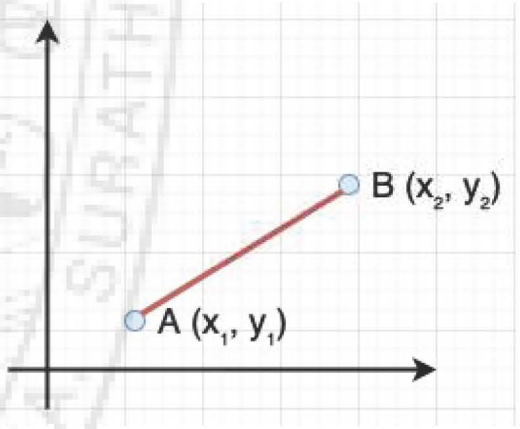
VSM: Evaluating Vector Closeness

Types of Distance Measures

Euclidean Distance:

- the length of the shortest segment connecting two points.

$$D(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2)}$$



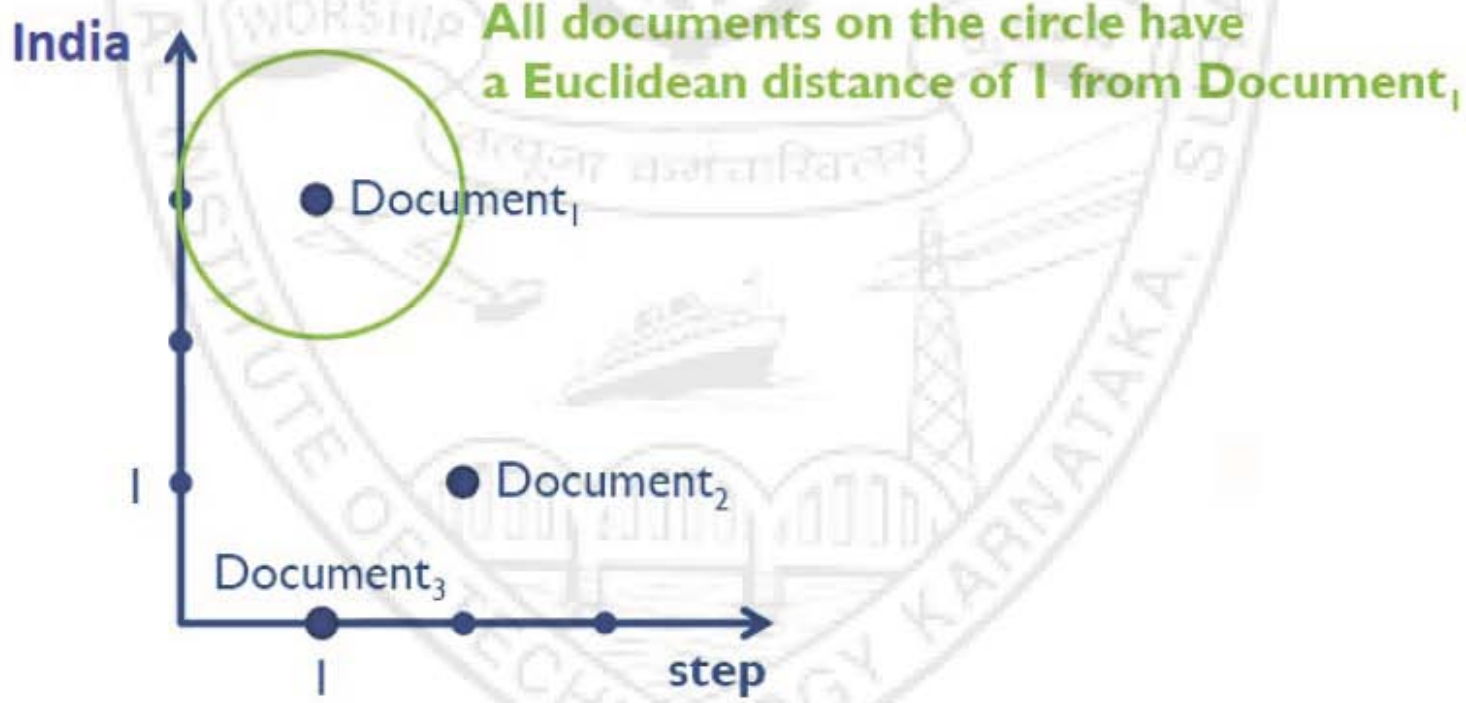
- where, $x = (x_1, x_2, \ldots x_i\}$ and $y = (y_1, y_2 \ldots y_i)$ are two vectors.

Disadvantages:

- Distances can get skewed depending on the units of features.
- Not suitable for higher dimensional data.

Distance measures (Example)

Euclidean Distance $d(x_1, \ldots, x_n; y_1, \ldots, y_n) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$



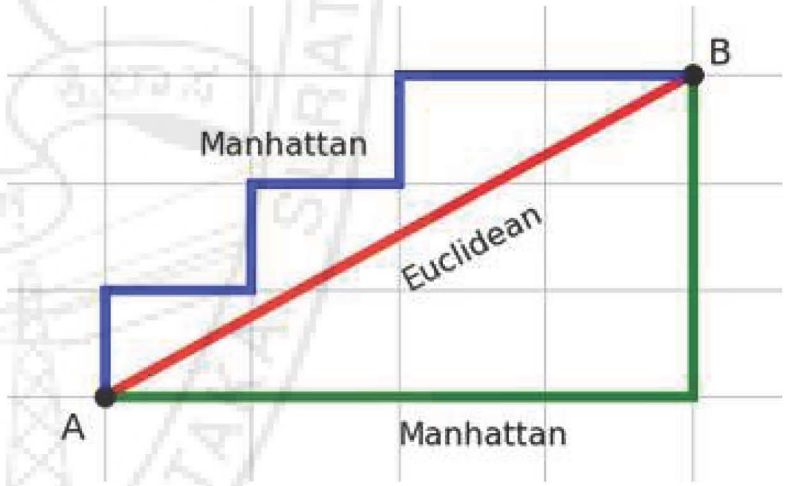
VSM: Evaluating Vector Closeness

Types of Distance Measures

Manhattan Distance (also called Cityblock distance):

- sum of the absolute differences between the two vectors.

$$D(x, y) = \sum_{i=1}^{k} |x_i - y_i|$$



- Details:
  - Works well for high-dimensional data.
  - Returns higher distance value than Euclidean distance since it does not measure the shortest path possible.

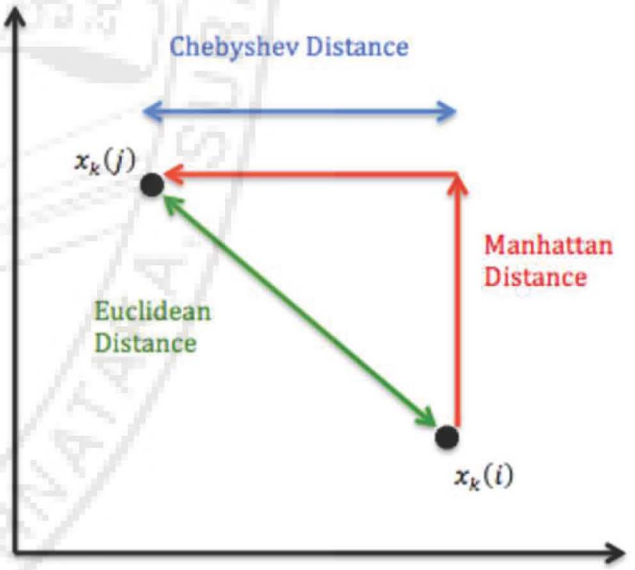
VSM: Evaluating Vector Closeness

Types of Distance Measures

Chebyshev Distance:

- the greatest of difference between two vectors along any coordinate dimension

$$D(x, y) = \max_i (|x_i - y_i|)$$



- Note:
  - Suitable for special cases where, magnitude in one dimension is to be measured.

VSM: Evaluating Vector Closeness

Types of Distance Measures

Minkowski Distance:

- generalized form of Euclidean and Manhattan Distance

$$D(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

- Note:
  - p=1 — Manhattan distance
  - p=2 — Euclidean distance
  - p=∞ — Chebyshev distance

VSM: Evaluating Vector Closeness

Types of Distance Measures

Jaccard Distance:

- measures how much word choice overlap there is between documents.

$$D(x, y) = 1 - \frac{|x \cap y|}{|y \cup x|}$$

- Note:
  - highly influenced by the size of the data.

VSM: Evaluating Vector Closeness

Types of Distance Measures

Sorenson Dice Distance:

- Measures the similarity between two documents.

$$D(x, y) = \frac{2\,|x \cap y|}{|x| + |y|}$$

This is for similarity

- Note:
  - Similar to Jaccard distance.

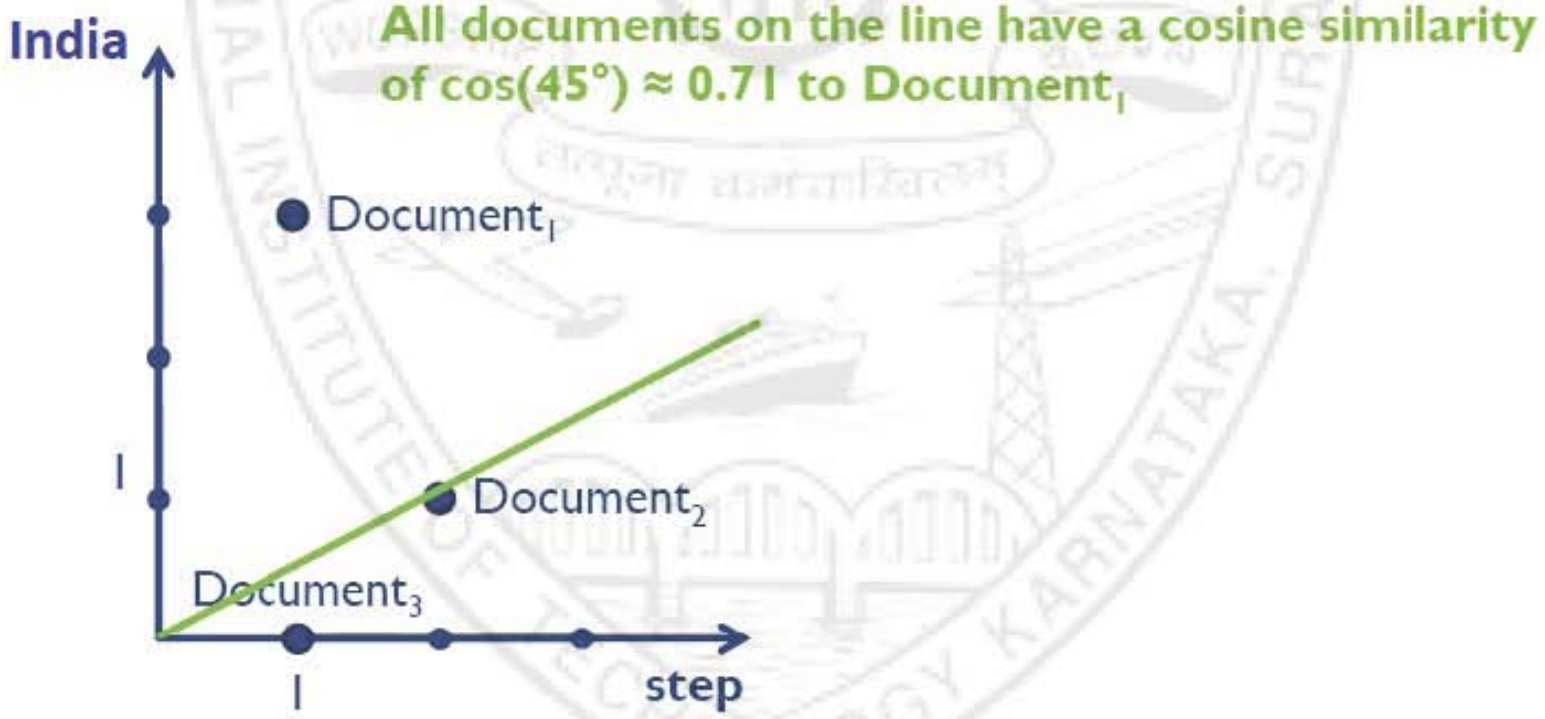
VSM: Evaluating Vector Closeness

Using "Similarity" for defining proximity

- A similarity measure on a set X is a function $\mathbf{s : X \times X \rightarrow [0, 1]}$ where
  - **s(x, y) = 1** → x and y are **maximally similar**
  - **s(x, y) = 0** → x and y are **maximally dissimilar**

Similarity Measures (Example)

- Cosine similarity in vector spaces $\mathbf{s(x,y) = \cos(\alpha)}$
  - α is the angle between these two vectors x and y



VSM: Evaluating Vector Closeness

- **Cosine Similarity Computation**

$$Cos - sim(q, d_i) = \cos(\alpha) = \frac{(d_i . q)}{|d_i| . |q|}$$

$$Cos - sim(q, d_i) = \cos(\alpha) = \frac{\sum_{j=1}^{t} w_{qj} w_{ij}}{\sqrt{\sum_{j=1}^{t} (w_{ij})^2 \sum_{j=1}^{t} (w_{qj})^2}}$$

Cosine Similarity Computation (Example)

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

$$D_2 = 3T_1 + 7T_2 + 1T_3$$

$$Q = 2T_3$$

$$Cos - sim(q, d_i) = \cos(\alpha) = \frac{\sum_{j=1}^{t} w_{qj} w_{ij}}{\sqrt{\sum_{j=1}^{t} (w_{ij})^2 \sum_{j=1}^{t} (w_{qj})^2}}$$

$$\text{Cos-sim}(D_1, Q) = (2*0+3*0+5*2) / \sqrt{(4+9+25)(0+0+4)} = 0.31$$

$$\text{Cos-sim}(D_2, Q) = (3*0+7*0+1*2) / \sqrt{(9+49+1)(0+0+4)} = 0.13$$

Cosine Similarity Computation (Example)

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

$$D_2 = 3T_1 + 7T_2 + 1T_3$$

$$Q = 2T_3$$

$$Cos - sim(q, d_i) = \cos(\alpha) = \frac{\sum_{j=1}^{t} w_{qj} w_{ij}}{\sqrt{\sum_{j=1}^{t} (w_{ij})^2 \sum_{j=1}^{t} (w_{qj})^2}}$$

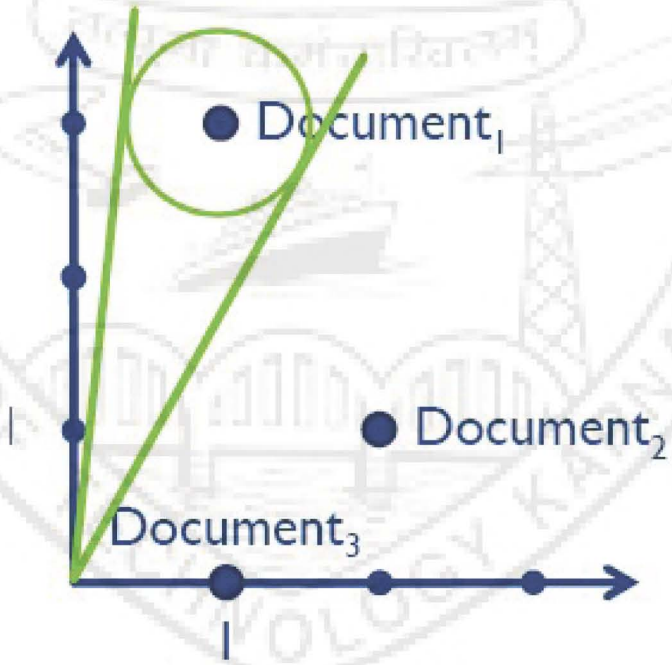
$$\text{Cos-sim}(D_1, Q) = (2*0+3*0+5*2) \,/\, \sqrt{(4+9+25)(0+0+4)} = 0.31$$

$$\text{Cos-sim}(D_2, Q) = (3*0+7*0+1*2) \,/\, \sqrt{(9+49+1)(0+0+4)} = 0.13$$

Insight: D_1 is nearly 2.5 times more relevant than D_2 w.r.t Q

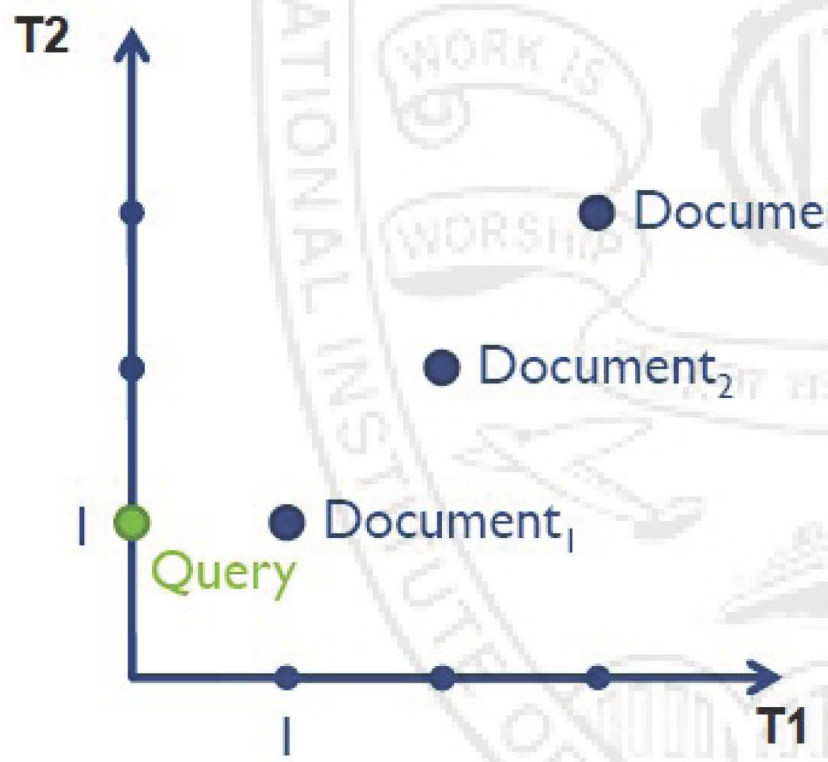
VSM: Evaluating Vector Closeness

- Choice always depends on the current application!
  - Different measures often behave alike, but not always …
- Example: *Low Euclidean distance implies high cosine similarity*
  - *the converse is not true*



VSM: Evaluating Vector Closeness

Need for Normalization



Question: When using Euclidean Distance, are shorter documents to be considered more similar to the query than longer ones???

VSM: Evaluating Vector Closeness

Need for Normalization

- Ways to normalize the vector representation of documents and queries

→ Divide each coordinate by the vector's length, i.e. **normalize to length 1:**

$$\frac{x}{\|x\|}$$

→ Divide each coordinate by the vector's largest coordinate

$$\frac{x}{\max_{i=1}^{n} x_i}$$

→ Divide each coordinate by the sum the vector's coordinates

$$\frac{x}{\sum_{i=1}^{n} x_i}$$

Weighting Terms in VSM

- *"Assign to each term a **weight** which is **proportional** to its **importance** both in the document and in the document collection."*

- Karen Spärck Jones

Sparck Jones, Karen. "A statistical interpretation of term specificity and its application in retrieval." *Journal of documentation* 28.1 (1972): 11-21.

Weighting Terms in VSM

- **Tf-idf term weighting scheme** $w_{i,j} = tf_{i,j} \times idf_i$

Term Frequency $tf_{i,j} = \begin{cases} 1 + \log f_{i,j} & \text{if } f_{i,j} > 0 \\ 0 & \text{otherwise} \end{cases}$

Inverse Document Frequency $idf_i = \log \frac{N}{n_i}$

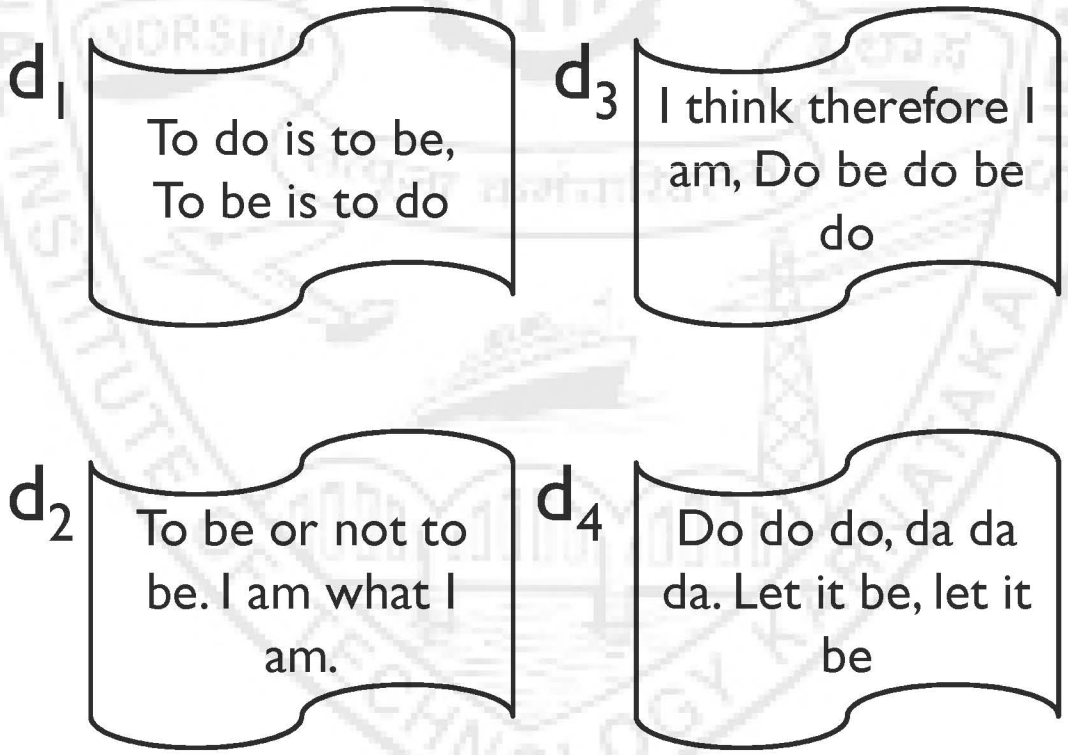
Where, N = size of the document corpus

n_i = Document frequency of a term k_i

$\mathbf{f_{i,j}}$ = frequency of a term k_i in document d_j

Vector Space Model (Example)

- Example *(Continued from previous class)*
  - Query q = "what I do"
  - Document collection



Vector Space Model (Process)

1) Preprocess the given corpus to generate the vocabulary.
2) Calculate the TF of each term in the vocabulary.
3) Calculate the IDF of each term in the vocabulary.
4) Compute the TF-IDF weight of each term in the vocabulary
5) Represent the corpus in vector space using the TF-IDF weights
   → Construct the document-term matrix
6) Represent the given query in the same vector space as the corpus
7) Calculate the Document Relevance using appropriate closeness measure
8) Generate the ranked list of relevant documents for the given query.

Vector Space Model (Example)

2. Calculate TF for the document corpus

- Term Frequency

$$tf_{i,j} = \begin{cases} 1 + \log f_{i,j} & \texttt{if } f_{i,j} > 0 \\ 0 & \texttt{otherwise} \end{cases}$$

To do is to be.
To be is to do.
d_1

To be or not to be.
I am what I am.
d_2

I think therefore I am.
Do be do be do.
d_3

Do do do, da da da.
Let it be, let it be.
d_4

Vocabulary		$tf_{i,1}$	$tf_{i,2}$	$tf_{i,3}$	$tf_{i,4}$
1	to	3	2	-	-
2	do	2	-	2.585	2.585
3	is	2	-	-	-
4	be	2	2	2	2
5	or	-	1	-	-
6	not	-	1	-	-
7	I	-	2	2	-
8	am	-	2	1	-
9	what	-	1	-	-
10	think	-	-	1	-
11	therefore	-	-	1	-
12	da	-	-	-	2.585
13	let	-	-	-	2
14	it	-	-	-	2

Vector Space Model (Example)

3. Calculate IDF for the document corpus

- IDF values of a sample collection $idf_i = \log \frac{N}{n_i}$

To do is to be.
To be is to do.

d_1

To be or not to be.
I am what I am.

d_2

I think therefore I am.
Do be do be do.

d_3

Do do do, da da da.
Let it be, let it be.

d_4

	term	n_i	$idf_i = \log(N/n_i)$
1	to	2	1
2	do	3	0.415
3	is	1	2
4	be	4	0
5	or	1	2
6	not	1	2
7	I	2	1
8	am	2	1
9	what	1	2
10	think	1	2
11	therefore	1	2
12	da	1	2
13	let	1	2
14	it	1	2

Vector Space Model (Example)

4. Compute TF-IDF weight of each term in the vocabulary

- Calculated TF-IDF for the example collection

To do is to be.
To be is to do.

d_1

To be or not to be.
I am what I am.

d_2

I think therefore I am.
Do be do be do.

d_3

Do do do, da da da.
Let it be, let it be.

d_4

	term	d_1	d_2	d_3	d_4
1	to	3	2	-	-
2	do	0.830	-	1.073	1.073
3	is	4	-	-	-
4	be	-	-	-	-
5	or	-	2	-	-
6	not	-	2	-	-
7	I	-	2	2	-
8	am	-	2	1	-
9	what	-	2	-	-
10	think	-	-	2	-
11	therefore	-	-	2	-
12	da	-	-	-	5.170
13	let	-	-	-	4
14	it	-	-	-	4

Vector Space Model (Example)

5. Represent the corpus in vector space using term weights

Process of representing the corpus in VSM

Docid/ terms	to	do	is	be	or	not	I	am	what	think	there fore	da	let	it
d_1														
d_2														
d_3														
d_4														

Vector Space Model (Example)

5. Represent the corpus in vector space using term weights

Process of representing the corpus in VSM

Docid/ terms	to	do	is	be	or	not	I	am	what	think	there fore	da	let	it
d_1	3	0.83	4	0	0	0	0	0	0	0	0	0	0	0
d_2	2	0	0	0	2	2	2	2	2	0	0	0	0	0
d_3	0	1.073	0	0	0	0	2	1	0	2	2	0	0	0
d_4	0	1.073	0	0	0	0	0	0	0	0	0	5.17	4	4

Vector Space Model (Example)

6. Represent the query in the same vector space

Process of representing the query in the same vector space

Docid/ terms	to	do	is	be	or	not	I	am	what	think	there fore	da	let	it
d_1	3	0.83	4	0	0	0	0	0	0	0	0	0	0	0
d_2	2	0	0	0	2	2	2	2	2	0	0	0	0	0
d_3	0	1.073	0	0	0	0	2	1	0	2	2	0	0	0
d_4	0	1.073	0	0	0	0	0	0	0	0	0	5.17	4	4
q														

Vector Space Model (Example)

6. Represent the query in the same vector space

Representing the query ="what I do" in VSM

$\text{Tf-idf}_{(\text{what, q})}$ =

$\text{Tf-idf}_{(\text{I, q})}$ =

$\text{Tf-idf}_{(\text{do, q})}$ =

Representing the query ="what I do" in VSM

take only corpus

$$\text{Tf-idf}_{(\text{what}, q)} = [1+\log(f_{i,j})] * [\log N/n_i] = [1+\log(1)] * [\log(4/1)] = 2$$

$$\text{Tf-idf}_{(\text{I}, q)} = [1+\log(1)] * [\log(4/2)] = 1$$

$$\text{Tf-idf}_{(\text{do}, q)} = [1+\log(1)] * [\log(4/3)] = 0.415$$

Vector Space Model (Example)

6. Represent the query in the same vector space

The corpus and the query ="what I do" represented in vector space

Docid/ terms	to	do	is	be	or	not	I	am	what	think	therefore	da	let	it
d_1	3	0.83	4	0	0	0	0	0	0	0	0	0	0	0
d_2	2	0	0	0	2	2	2	2	2	0	0	0	0	0
d_3	0	1.073	0	0	0	0	2	1	0	2	2	0	0	0
d_4	0	1.073	0	0	0	0	0	0	0	0	0	5.17	4	4
q	**0**	**0.415**	**0**	**0**	**0**	**0**	**1**	**0**	**2**	**0**	**0**	**0**	**0**	**0**

Vector Space Model (Example)

7. Calculate document relevance

Document Relevance calculation using Cosine Similarity

$$Cos - sim(q, d_i) = \cos(\alpha) = \frac{\sum_{j=1}^{t} w_{qj} w_{ij}}{\sqrt{\sum_{j=1}^{t} (w_{ij})^2 \sum_{j=1}^{t} (w_{qj})^2}}$$

common words used here

Cos-sim (q, d_1) = (3*0+0.83*0.415+4*0+0*1/ (4.205*2.274) = 0.036

Cos-sim(q, d_2) = 0.539

Cos-sim(q, d_3) = 0.285

Cos-sim(q, d_4) = 0.025

Vector Space Model (Example)

8. Generate ranked list of relevant documents

Document Relevance calculation using Cosine Similarity

$$Cos - sim(q, d_i) = \cos(\alpha) = \frac{\sum_{j=1}^{t} w_{qj} w_{ij}}{\sqrt{\sum_{j=1}^{t} (w_{ij})^2 \sum_{j=1}^{t} (w_{qj})^2}}$$

$$\text{Cos-sim}\ (q, d_1) = 0.036$$

$$\text{Cos-sim}(q, d_2) = 0.539$$

$$\text{Cos-sim}(q, d_3) = 0.285$$

$$\text{Cos-sim}(q, d_4) = 0.025$$

Final Ranking → d2 > d3 > d1 > d4

Vector Space Model - Summary

- **Pros**
  - Highly customizable and adaptable to specific collections:
    - Term weighting schemes
    - Normalization schemes
    - Distance/similarity functions
  - Ranking
  - Relevance feedback possible

Vector Space Model - Summary

- **Cons**
  - Missing syntactic information
    - phrase structure
    - word order
  - Assumption of term independence
    - e.g. ignores synonomy
  - Missing semantic information
    - e.g. word sense

Further reading...

- Sparck Jones, Karen. "A statistical interpretation of term specificity and its application in retrieval." *Journal of documentation* 28.1 (1972): 11-21.

- Salton, Gerard, Anita Wong, and Chung-Shu Yang. "A vector space model for automatic indexing." *Communications of the ACM* 18.11 (1975): 613-620.

- Salton, Gerard, and Robert Kenneth Waldstein. "Term relevance weights in on-line information retrieval." *Information Processing & Management* 14.1 (1978): 29-35.