(Recall) Bag of Words Representation

What is a bag of word representation?

- Simple strategy for representing documents
- Count how many times each term occurs
  - Binary mode uses only 0 & 1
- A 'term' is any lexical item that you chose such as:
  - A word (delimited by 'white space' or punctuation)
  - Some conflated 'root form' of each word (e.g. a stem)
  - An n-gram (a sequence of any consecutive n chars)
- Doesn't consider the ordering of words in a document
  - John is quicker than Mary and Mary is quicker than John have the same representation
  - This could be a set back: **positional information** allows to distinguish these 2 docs
- For now: Bag of Words Model (BoW)

3

Vector Space Model

LET'S LOOK AT THIS PROCESS DIFFERENTLY ….

Document Vectors
One location for each word

	diet	film	fur	galaxy	heat	h'wood	nova	role
A				5	3		10	

"Nova" occurs 10 times in text A
"Galaxy" occurs 5 times in text A
"Heat" occurs 3 times in text A
(Blank means 0 occurrences.)

Document Vectors
One location for each word

"Hollywood" occurs 7 times in text I
"Film" occurs 5 times in text I
"Diet" occurs 1 time in text I
"Fur" occurs 3 times in text I

	diet	film	fur	galaxy	heat	h'wood	nova	role
I	1	5	3			7		

Document Vectors
One vector for each document

Document ids

	diet	film	fur	galaxy	heat	h'wood	nova	role
A				5	3		10	
B	5	10						
C				10	8	7		
D				9	10	5		
E							10	10
F							9	10
G	5	7			9			
H		6	10	2	8			
I	1	5	3			7		

7

Vector Space Model

- Documents are also treated as a "bag" of words or terms
- Each document is represented as a vector in a *t-dimensional* vector space (*t* is the number of index terms)
- Each term weight is computed based on some variations of TF or TF-IDF scheme

8

TF-IDF Vectors

Document ids

	diet	film	fur	galaxy	heat	h'wood	nova	role
A				8	.5		.6	
B	4	1						
C				3	4	2.8		
D				2.7	5	2		
E							9	1.5
F							8.1	1.5
G	4	2.8			3.6			
H		.6	4		1			
I	2.1	2.5	.9				.45	

Sparse Matrix

More Formally ….

- Documents and queries are represented by vectors of term weights
- A collection is represented by a matrix of term weights

$D_i = (d_{i1}, d_{i2}, \ldots, d_{it}) \quad Q = (q_1, q_2, \ldots, q_t)$

$$\begin{array}{lcccc} & Term_1 & Term_2 & \ldots & Term_t \\ Doc_1 & d_{11} & d_{12} & \ldots & d_{1t} \\ Doc_2 & d_{21} & d_{22} & \ldots & d_{2t} \\ \vdots & \vdots & & & \\ Doc_n & d_{n1} & d_{n2} & \ldots & d_{nt} \end{array}$$

So, we have docs represented as vectors

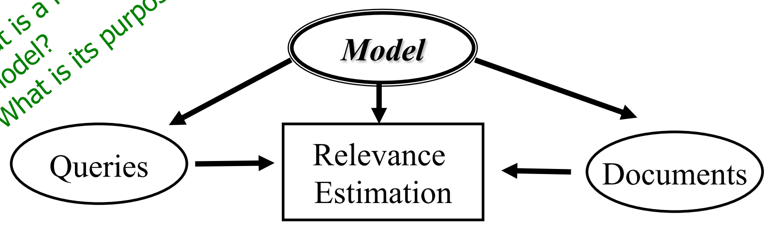
How can we use this in retrieval?

t is the number of index terms (words, stems, etc)

10

What is a retrieval model?
What is its purpose?

Retrieval Model



An IR Model defines
- a model for document representation
- a model for query representation
- a mechanism for estimating the relevance of a document for a given query

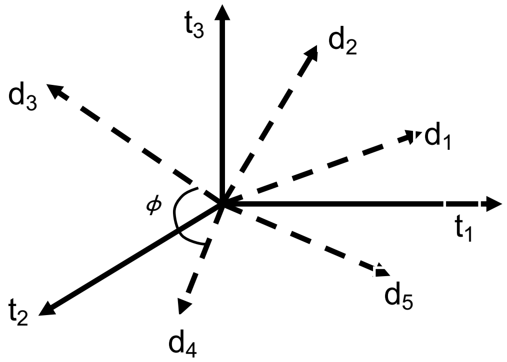
Progress in retrieval models has corresponded with improvements in effectiveness

Retrieval in Vector Space Model

- Vector space model represents both query and documents using term sets (term vectors)
- Documents and queries are represented in a high dimensional space (Bag of Words)
  - Each dimension of the space corresponds to a term in the document collection (*t-dimensional vector space*)
- Relevance Estimation is performed by identifying documents **similar** to the query
  - Relevance of $\mathbf{d}_i$ to $\mathbf{q}$: Compare the **similarity** of query $\mathbf{q}$ and document $\mathbf{d}_i$

12

Geometrically: Vector Space Model

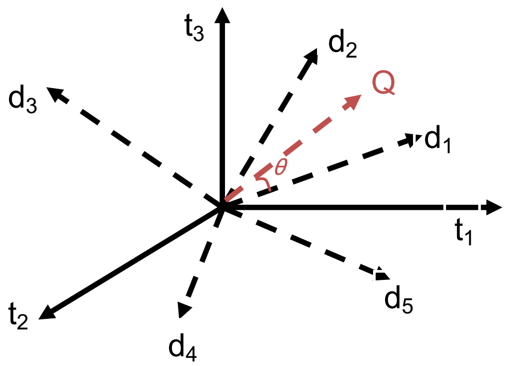


Assumption: Documents that are "close together" in vector space "talk about" the same things

NB: **3D diagrams useful, but can be misleading for high-dimensional space**

13

Geometrically: Vector Space Model



Assumption: Documents that are "close together" in vector space "talk about" the same things

Therefore, retrieve documents based on **how close the document is to the query** (i.e., similarity ~ "closeness")

14

Vector Space

- $X = (t_1, t_2, \ldots, t_t)$
  - The number t_i is called the **i**-th component of the vector
  - **Magnitude:** is defined by the square root of the sum of the squares of the components
    - that is, $\sqrt{\sum t_i^2}$
  - If $||X|| = 1$ then X is a **unit vector**
    - *Concept of length normalization*

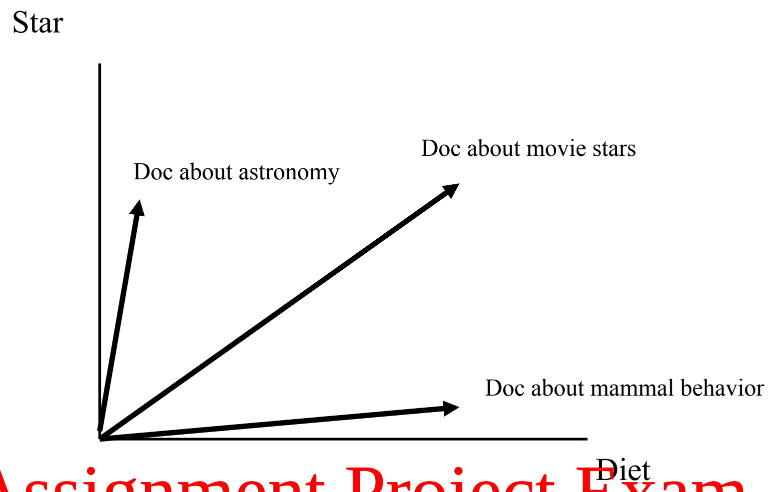
15

Summary: Document Vectors

- Documents are represented as "bags of words"
- Represented as vectors when used computationally
  - A vector is like an array of floating point
  - Has direction and magnitude
  - Each vector holds a (unique) place for **every term** in the collection
  - Therefore, most vectors are sparse

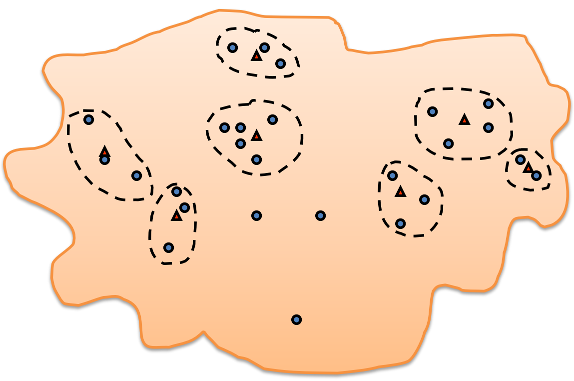
16

Plotting the Vectors … & Intuition



Vector Space Intuition

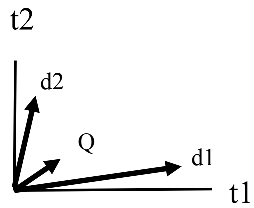
- Library
  - Books from a domain are organised at the same place/ shelf/ nearby shelves
  - Human organisation - librarian



What is the intuition behind The vector-space model?

Vector Space Model

- The relevant documents for a query are expected to be those represented by the vectors closest to the query



- Documents ranked by distance between points representing query and documents
  - ***Similarity*** measure more common than a distance or *dissimilarity* measure
  - e.g.

$$Cosine(D_i, Q) = \frac{\sum_{j=1}^{t} d_{ij} \cdot q_j}{\sqrt{\sum_{j=1}^{t} d_{ij}^2 \cdot \sum_{j=1}^{t} q_j^2}}$$

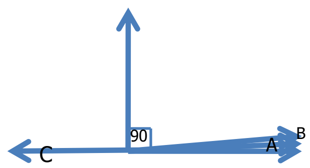
19

Cosine Measure

In IR we consider only the similarity range from 0 to 1
Why? Why not -1 to 1?

- It measures cosine of the angle between the vectors
- Cosine ranges from 1 for vectors pointing in the same direction over zero for orthogonal vectors and -1 for vectors pointing in opposite directions
- If Cosine is applied to normalised (unit) vectors it gives the same ranking as Euclidean distance does

Cos 0′ = 1 Cos 90′ = 0
Cos 180′ = -1



20

Similarity Calculation

– Consider two documents D_1, D_2 and a query Q
  - $D_1 = (0.5, 0.8, 0.3)$, $D_2 = (0.9, 0.4, 0.2)$, $Q = (1.5, 1.0, 0)$

WITHELD

How could we implement a cosine similarity-based measure using inverted index?

What role the denominator plays?

Algorithm (Reminder)

For each document I, Score(I) =0; I = 1 to N

For each query term t_k

– Search the vocabulary list
– Pull out the postings list
– For each document J in the list,
  - Score(J) =Score(J) + w_{kj}

Example

- D1 = (T1 => 12 ,T2=> 23 , T3=>3)
- D2 = (T1 => 3 , T2 => 2 , T3 => 1)
- Q = (T1 => 0 ,T2=> 0, T3=>2)

- Sim(D1,Q) = 12*0 + 23*0 + 3*2 =6
- Sim(D2,Q) = 3*0+3*0+1*2 = 2

sim 为什么没把分母考虑在内?.？

If you are using an inverted index?

3*2=6

1*2 = 2

of Document length?

Matching Coefficient (Coordination Level)

- Simply counts the number of dimensions on which both vectors are non-zero
- $|X \cap Y| \equiv \sum x_i * y_i$

Is this familiar?

- Number of shared index terms (binary vectors)

- Does not take into account the sizes of the vectors

Some Problems …

- Normalisation …
- Consider a single word query and a single word document (In Binary mode...)
  - If that matches
    - Coefficient is 1
- Same query against a thousand word document
  - If that matches
    - Coefficient is 1

Justify the need for vector length normalization

Dice Coefficient

- $2 \; |X \cap Y| \, / \, (|X| + |Y|)$
- Normalises for length by dividing by the total number of non-zero entries.
- We multiply by 2 so that we get a measure that ranges from 0 to 1.0

What is a dice coefficient?

Query Term Weighting

- Boolean representation
  - Just have a weight of zero or 1
- Short queries
  - Typical of web searches
  - Multiple keyword occurrences are rare
    - $W_{kq} = idf_k$
- Long queries
  - Result of **relevance feedback** (will talk about it later)
    - $W_{kq} = f_{kq} \cdot idf_k$

Discuss three query term weighting strategies!

27

Advantages and Disadvantages of a Vector-space Model

Advantages

- Simple **geometric interpretation** of retrieval readily comprehensible to non-specialist and a uniform basis for wide range of operations
- Easy to compute measure (any similarity measure can be used)
- Easy to adapt to various weighting schemes
- Provision for ***ranked output***

Disadvantages

- High dimensionality
- Term independence assumption
- Adhoc similarity metric: Cosine, Dice, etc. (which one to use?)
- Adhoc term weighting (not theoretically founded)
- No guidance on when to stop ranking

28