Plan for Today's Lecture(s)

- Motivating The Vector Model
- The Vector Model
- Term Weighting
- Similarity Calculation
- Limitations of the Vector Model

UNIVERSITY OF CALIFORNIA, BERKELEY
SCHOOL OF INFORMATION

INFO 202
"Information Organization & Retrieval"
Fall 2013

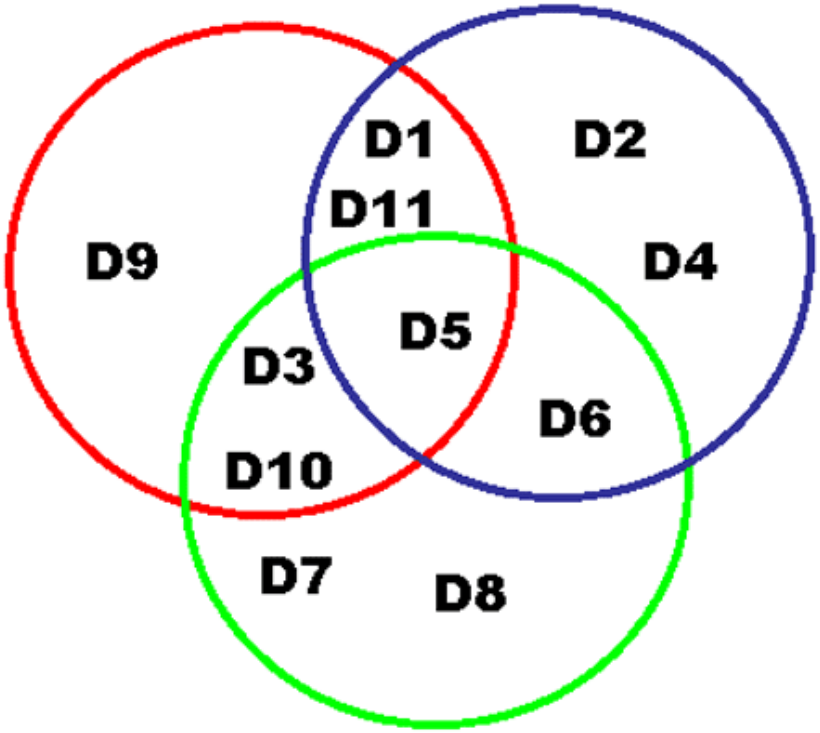
Robert J. Glushko
glushko@berkeley.edu
@rjglushko

12 November 2013
Lecture 22.1 – Motivating the Vector Model

The Boolean Model

- The Boolean model of IR represents documents and queries as sets of index terms
- Term frequency is not represented in any way; terms are either present or absent
- This enables comparisons between queries and document collections using set theory operations with Boolean algebra, making relevance a binary decision

Example of Boolean Representation



docs	*t1*	*t2*	*t3*
D1	1	0	1
D2	1	0	0
D3	0	1	1
D4	1	0	0
D5	1	1	1
D6	1	1	0
D7	0	1	0
D8	0	1	0
D9	0	0	1
D10	0	1	1
D11	1	0	1

Relevance in the Boolean Model

- Synonymy and polysemy mean that the presence or absence of any specific term is an inadequate measure of relevance
  - Synonymy means that relevant documents are missed
  - Polysemy means that irrelevant documents are considered relevant

Relevance in the Boolean Model

- Because terms are either present or absent, a document is either relevant or not relevant
- Retrieved documents might be ordered (chronologically?) but not by relevance because there is logically no way to calculate it
- But this is clearly flawed -- if a query is "a and b" the Boolean model retrieves only those documents that contain both of them, and treats documents that contain only a or only b as equally irrelevant

Relevance in other IR Models

- Other IR models rank the retrieved documents in terms of their relevance to the query
- This requires more refined methods of representing what documents are about
- This enables more continuous methods of assessing relevance rather than all or none

Motivating Term Weighting

- The Boolean model represents documents as a set of index terms that are either present or absent
- This binary notion doesn't fit our intuition that terms differ in how much they suggest what the document is about
- We will capture this notion by assigning weights to each term in the index

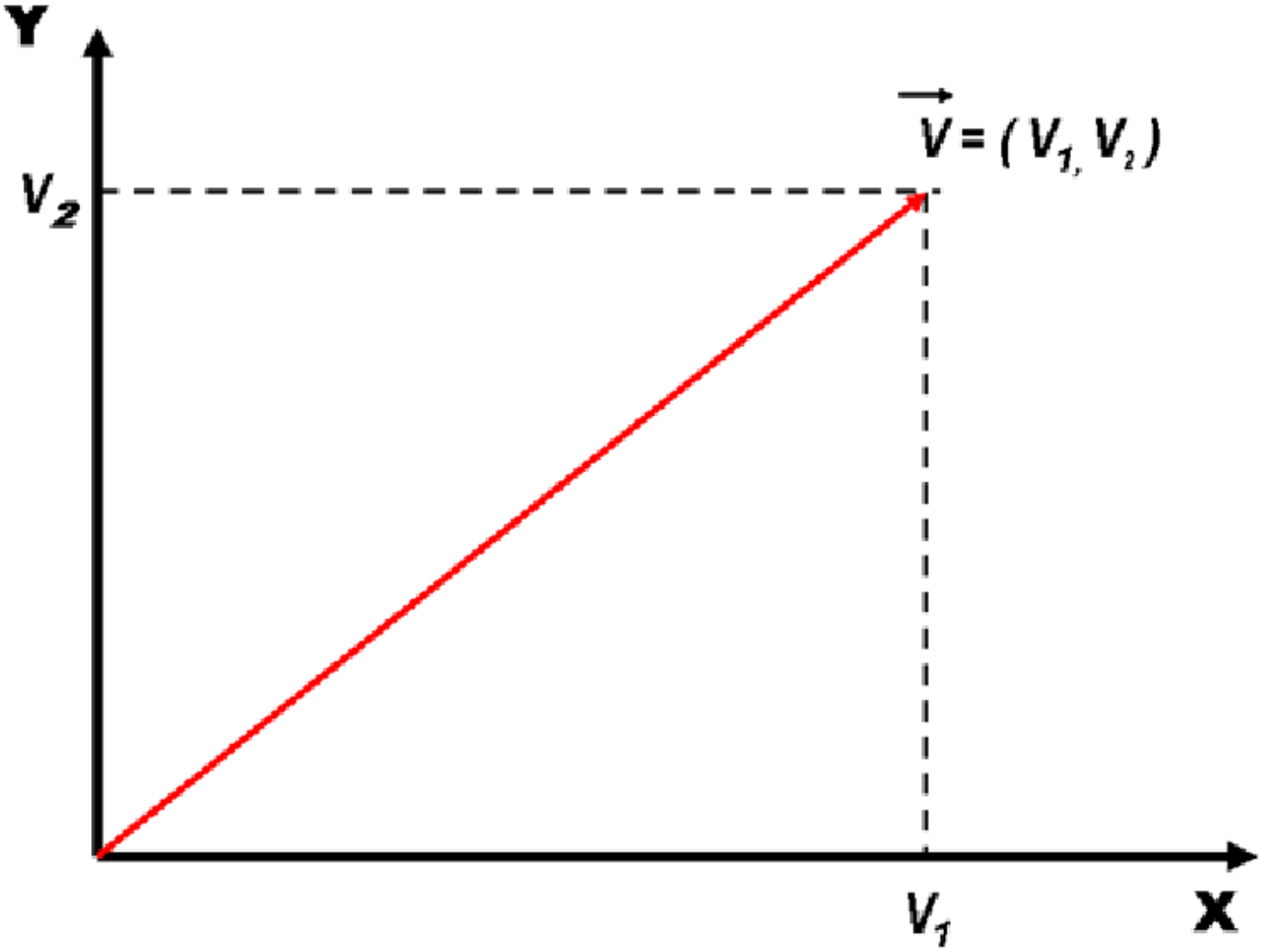
UNIVERSITY OF CALIFORNIA, BERKELEY
SCHOOL OF INFORMATION

INFO 202
"Information Organization & Retrieval"
Fall 2013

Robert J. Glushko
glushko@berkeley.edu
@rjglushko

12 November 2013
Lecture 22.2 – The Vector Model

Vectors



Vectors

- Vectors are an abstract way to think about a list of numbers
- Any point in a vector space can be represented as a list of numbers called "coordinates" which represent values on the "axes" or "basis vectors" of the space

Vectors

- Adding and multiplying vectors gives us a way to represent a continuous space in any number of dimensions
- We can multiply a coordinate value in a vector to "scale" its length on a particular basic vector to "weight" that value (or axis)

Overview of Vector Model

- Documents and queries are represented in the same way as word or term vectors
- Each dimension is the count of occurrences for a term
- Term weights can capture term counts within a document or the importance of the term in discriminating the document in the collection

Overview of Vector Model

- Vector algebra provides a model for computing similarity between queries and documents and between documents because of assumption that "closeness in space" means "closeness in meaning"
- There is no representation of word order (sentence structure) in the vector - it is just a BAG OF WORDS

An Important Note on Terminology

- WARNING: A lot of IR literature uses Frequency to mean Count
  - Term Frequency is defined to mean "the number of occurrences of a term in a document"
  - ... even though to actually make it a frequency the count should be divided by some measure of the document's length

Document x Term Matrix

- We can represent for each document the frequency of the words (or terms created by stemming morphologically related words) that it contains

ID	nova	galaxy	heat	h'wood	film	role	diet	fur
A	10	5	3					
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				7	5		1	3

Document Vector

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

"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 Vector

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

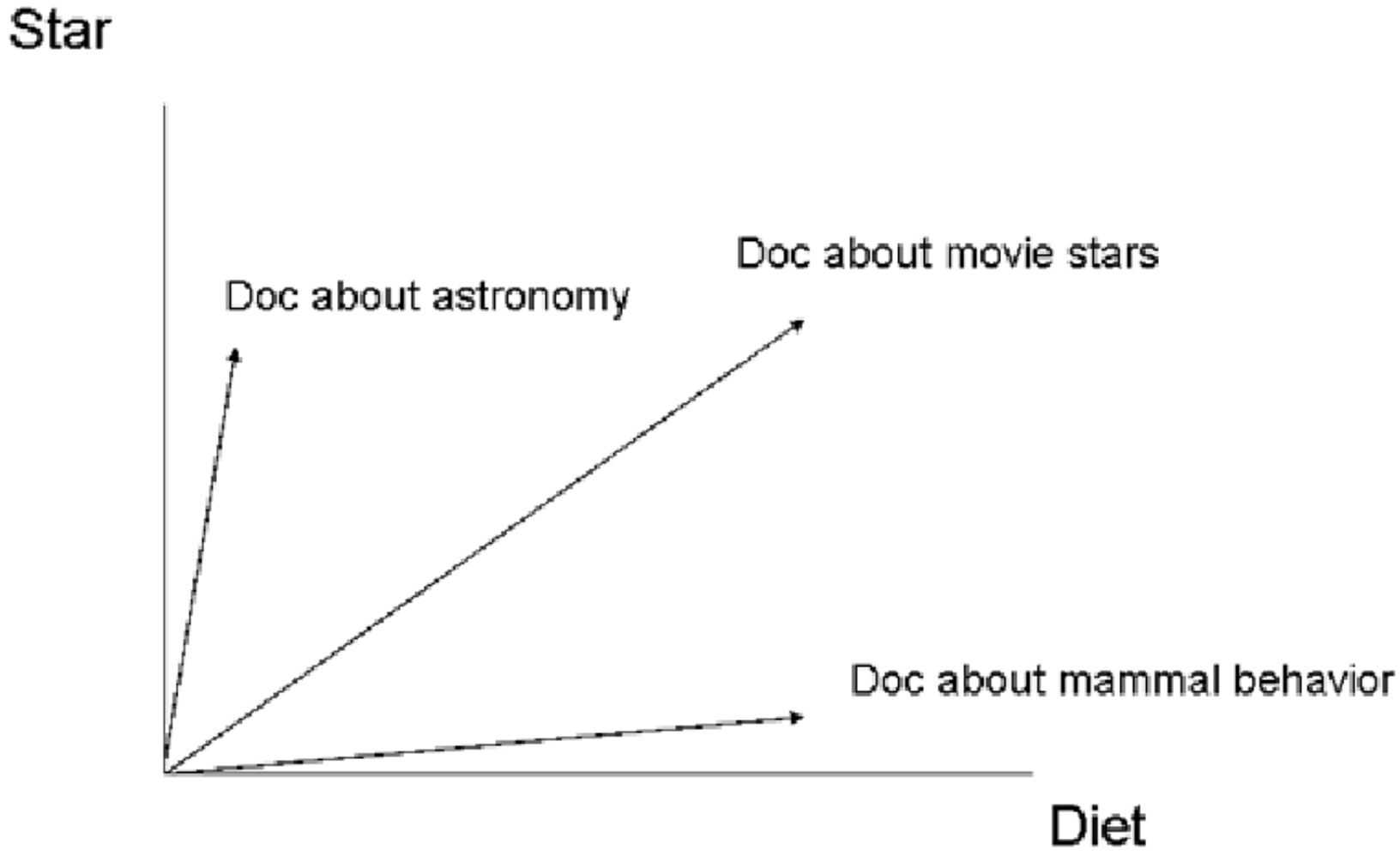
"Hollywood" occurs 7 times in text I
"Film" occurs 5 times in text I
"Diet" occurs 1 time in text I
"Fur" occurs 3 times in text I

Word (or Term) Vectors

- We can use this same matrix to represent a word / term as a vector whose coordinates measure how much it indicates the concept or context of a document

ID	nova	heat	film	diet
A	10	3		
B	5			
C			8	
D			10	
E				10
F				9
G	5	7		
H		10	8	
I			5	1

Documents in Term Space – 2D



UNIVERSITY OF CALIFORNIA, BERKELEY
SCHOOL OF INFORMATION

INFO 202
"Information Organization & Retrieval"
Fall 2013

Robert J. Glushko
glushko@berkeley.edu
@rjglushko

12 November 2013
Lecture 22.3 – Term Weighting

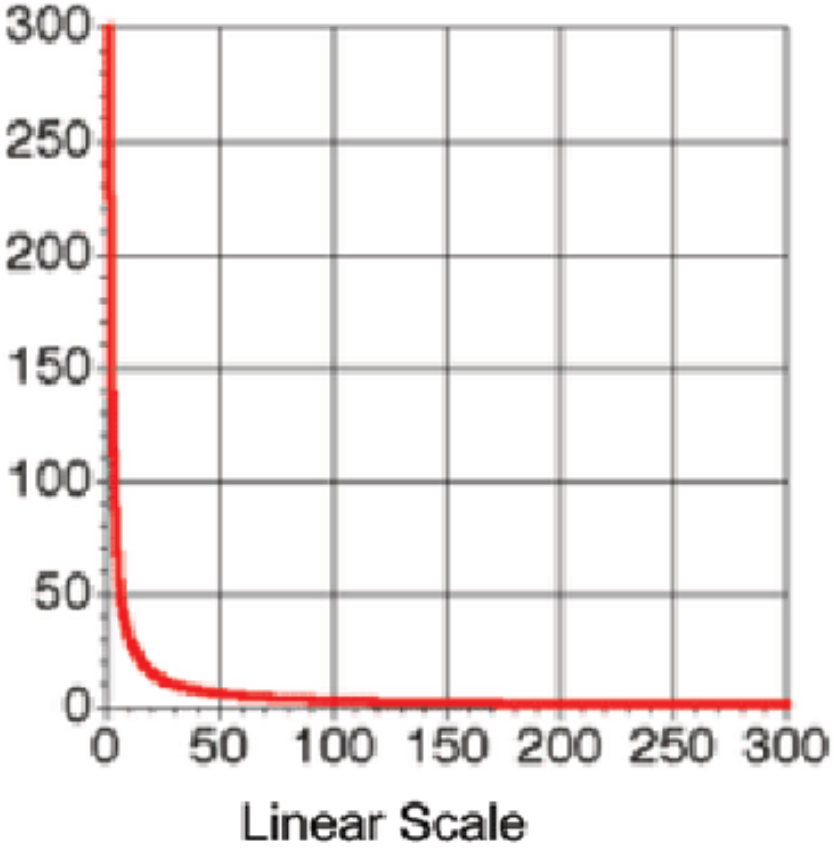
The Zipf Distribution

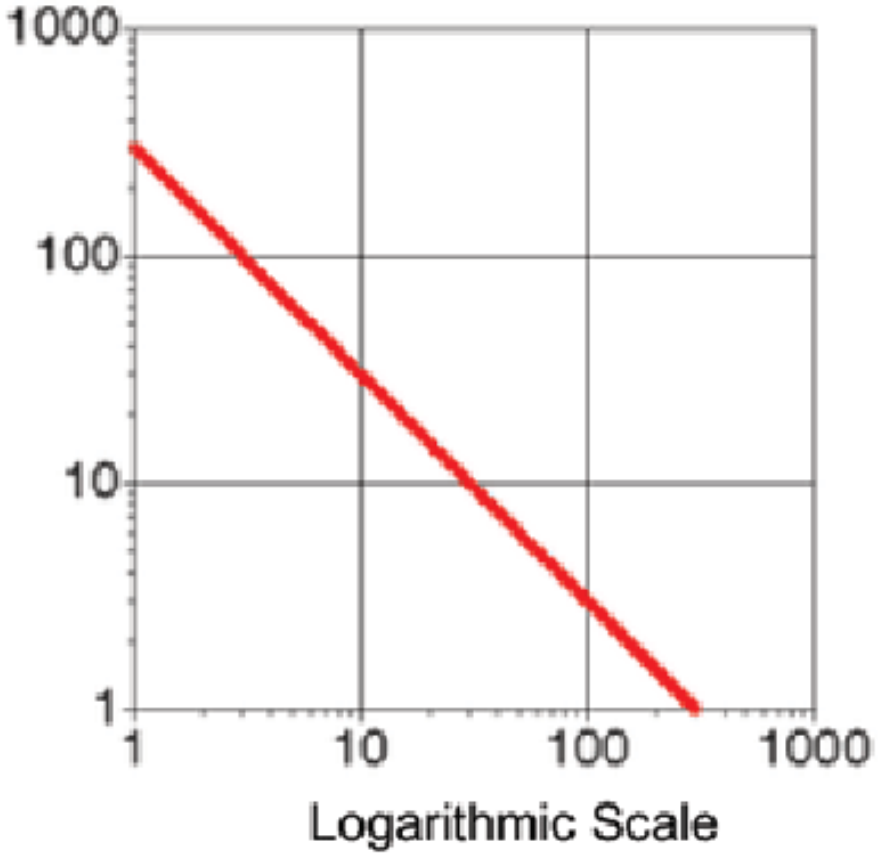
- In any natural language, we can observe that:
  - A few items occur very frequently
  - A medium number of elements have medium frequency
  - Very many elements occur very infrequently (the "long tail“)

The Zipf Distribution

- An approximate model of this distribution is the Zipf Distribution, which says that the frequency of the i-th most frequent word is 1/(i^a) times that of the most frequent word
- Because of the huge range in frequency and the long tail, the distribution is often drawn on a log-log scale

Zipf Distribution – Linear vs. Log Plotting



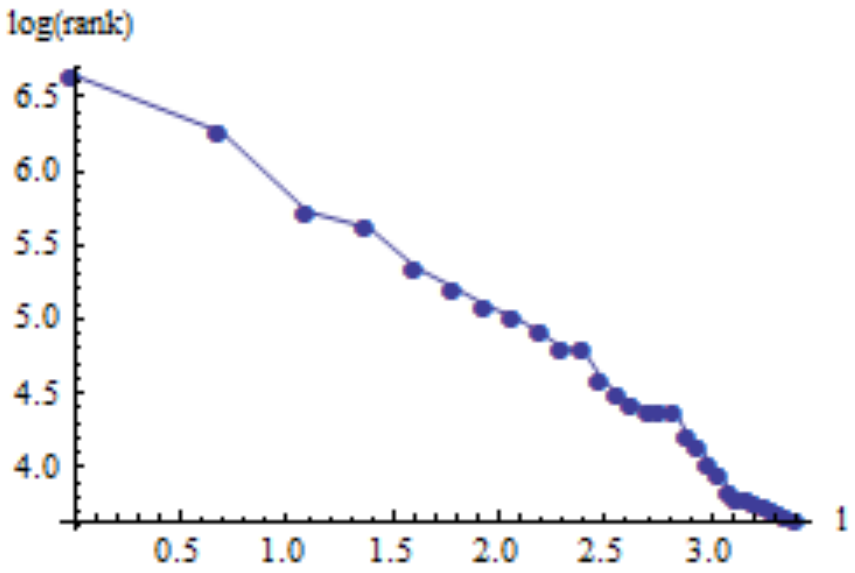


Zipf Distribution – US Constitution

frequencies

the 766	of 523	shall 306
and 275	to 210	be 183
or 162	in 150	states 135
by 122	president 120	a 99
united 89	for 84	state 80
any 79	congress 79	as 67
have 63	section 56	such 52
may 46	not 44	which 44
all 43	no 42	on 41
from 40	law 39	amendment 38

frequency plot



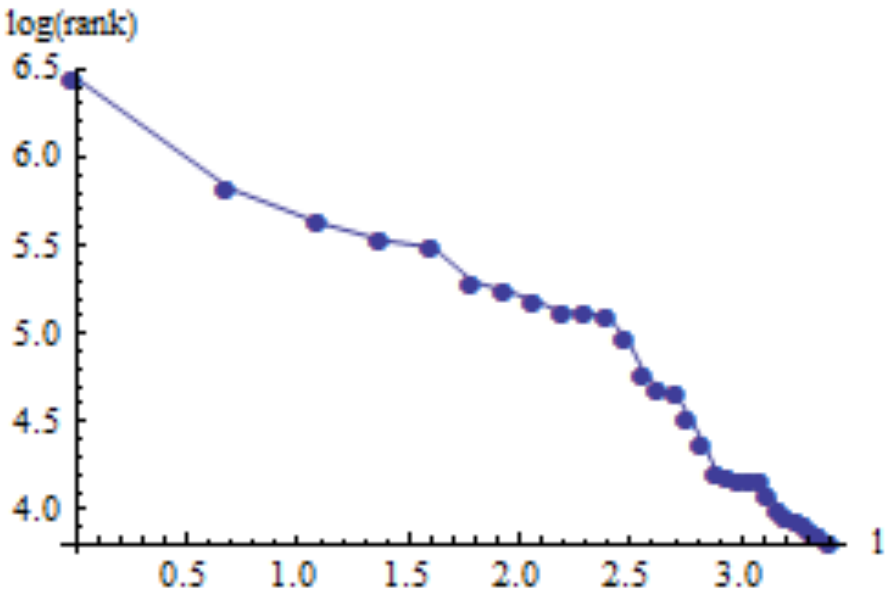
http://demonstrations.wolfram.com/ZipfsLawAppliedToWordAndLetterFrequencies/

Zipf Distribution – Alice in Wonderland

frequencies

the 632	and 338	a 278
to 252	she 242	of 199
it 189	i 178	was 167
alice 167	in 163	said 144
you 118	her 108	that 105
as 91	at 79	with 67
s 66	had 65	all 64
on 64	little 59	out 54
down 52	this 51	t 50
for 48	but 47	they 45

frequency plot

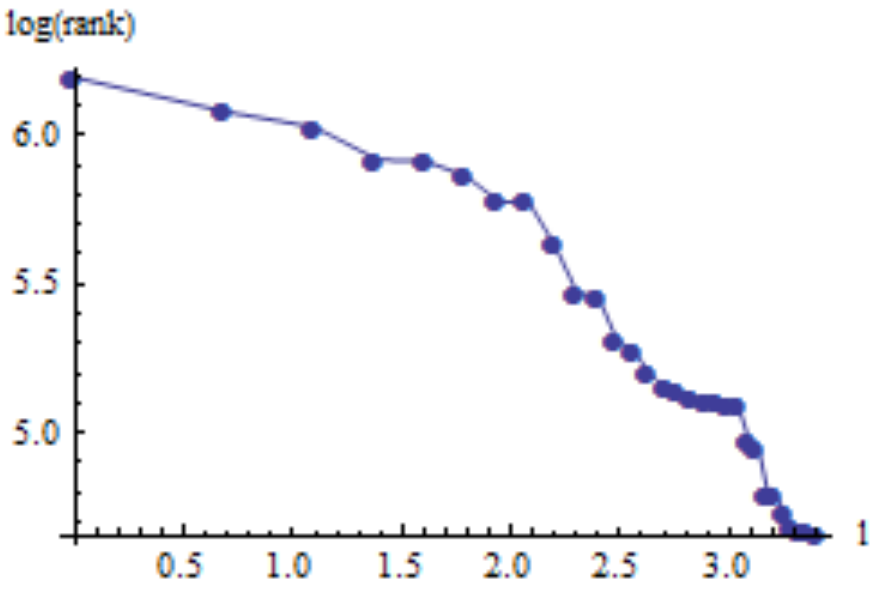


Zipf Distribution – Shakespeare's Sonnets

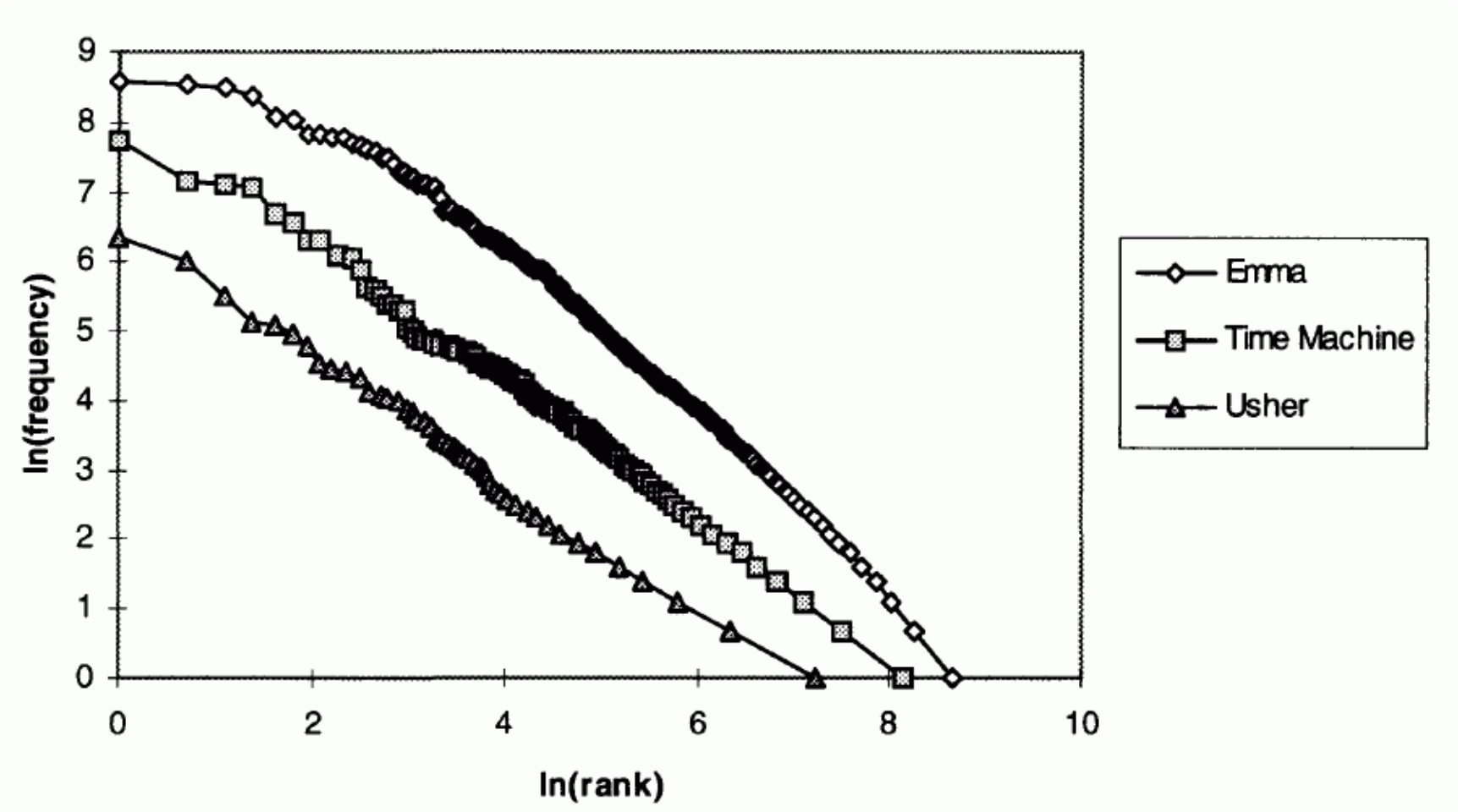
frequencies

and 490	the 437	to 415
my 372	of 370	i 352
in 323	that 323	thy 280
d 238	thou 235	s 203
love 195	with 181	for 172
is 170	not 167	me 164
a 164	but 163	thee 162
so 145	be 141	as 121
all 120	you 113	which 108
his 107	when 106	this 105

frequency plot



Zipf for 3 Novels on Same Plot



Resolving Power

- The term distribution lets us categorize terms on the basis of how well they discriminate between the documents in the collection.
- The value of a term varies inversely with the number of documents in which it occurs
- The most informative words are those that occur infrequently but when they occur they occur in clusters

Resolving Power and Term Weighting

- Terms that appear in every document have no resolving power because including them retrieves every document
- Terms that appear very infrequently have great resolving power, but don't turn out to be very useful because they so rare that most people will never use in queries

Resolving Power and Term Weighting

- So the most useful terms are those that are of intermediate frequency but which tend to occur in clusters, so most of their occurrences are in a small number of documents in the collection

Weighting Using Term Frequency

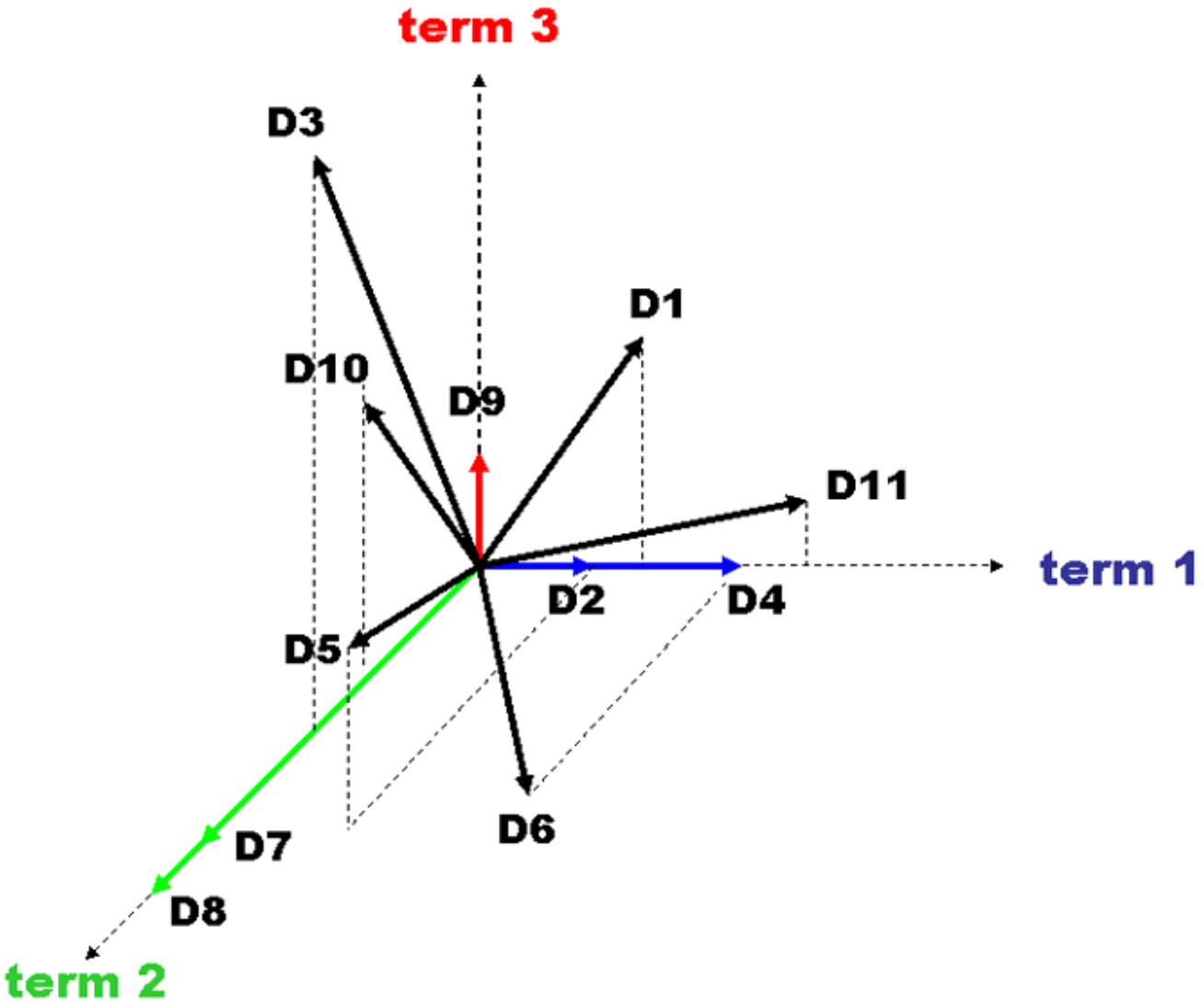
docs	*t1*	*t2*	*t3*
D1	2	0	3
D2	1	0	0
D3	0	4	7
D4	3	0	0
D5	1	6	3
D6	3	5	0
D7	0	8	0
D8	0	10	0
D9	0	0	1
D10	0	3	5
D11	4	0	1

Instead of a binary "yes" or "no" for the presence of a term in a document, we can use the frequency of the term as a weight in the document representation

This seems natural -- authors repeat words in documents to emphasize them -- so frequency is a measure of "aboutness"

But doesn't this create a bias toward long documents that are verbose?

Term Frequency Weighted Vectors



Term Resolving Power

docs	t1	t2	t3	t4	t5
D1	2	0	3	0	5
D2	1	0	0	0	7
D3	0	4	7	0	6
D4	3	0	0	0	6
D5	1	6	3	0	8
D6	3	5	0	0	7
D7	0	8	0	0	7
D8	0	10	0	0	7
D9	0	0	1	0	9
D10	0	3	5	1	7
D11	4	0	1	0	12

Inverse Document Frequency

- We need a way to penalize the words that are too frequent so they don't get in the way of the terms that have greater resolving power
- We will replace the actual term frequency in the vector with one that we calculate using some weighting function
- “Inverse document frequency” is the weighting function that “penalizes” the too frequent words

Inverse Document Frequency

N = total number of documents in a collection

df_t = document frequency for term *t*
(the number of documents that contain *t*)

idf_t = inverse document frequency

$= \log (N / df_t)$

IDF Examples

N = collection size = 10000

tf	idf
1	log(10000/1) = 4
10	log(10000/10) = 3
50	log(10000/50) = 2.301
100	log(10000/100) = 2
1000	log(10000/1000) = 1
5000	log(10000/5000) = .301
10000	log(10000/10000) = 0

Rationale for TF x IDF

- The IDF calculation gives us a measure of how important the term overall is in the collection
- But we need to do another calculation to get at the "clustering" intuition that the terms that do the best job finding relevant documents are those that occur in clumps to indicate the aboutness of a document
- So we will multiply IDF by TF (i.e. the count) within each document to compute the term weights

Calculating Term Weights

tf_{td} = term frequency for term *t* in document *d*

idf_t = inverse document frequency of term *t* in the collection

$tf\text{-}idf_{td} = tf_{td} \times idf_t$

(sometimes called w_{td}, the weight of *t* in *d*)

Calculated Term Weights for TF x IDF

- Term weights are highest for terms that occur many times within a small number of documents (high IDF)
- They are lower when the term occurs fewer times in a document (low TF), or occurs in many documents (low IDF)
- They are lowest when the term occurs in virtually all documents (almost 0 IDF)

UNIVERSITY OF CALIFORNIA, BERKELEY
SCHOOL OF INFORMATION

INFO 202
"Information Organization & Retrieval"
Fall 2013

Robert J. Glushko
glushko@berkeley.edu
@rjglushko

12 November 2013
Lecture 22.4 – Similarity and Relevance in the Vector Model

Calculating Similarity in Vector Models

- The similarity between a query and a document, or between two documents, is defined as the "inner product" of their vectors
- The inner product of two vectors is the sum of the products of their overlapping terms (so similarity increases when they have more terms in common)

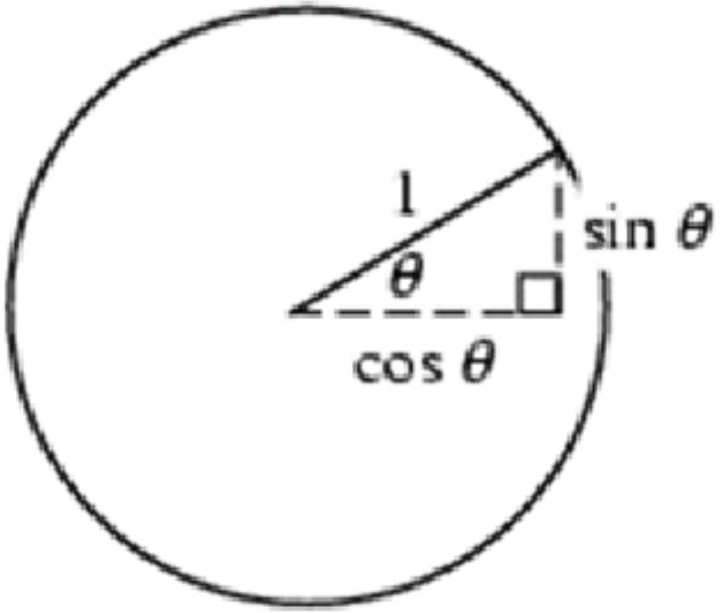
v1 = (3, 8, 4) v2 = (2, 3, 1)

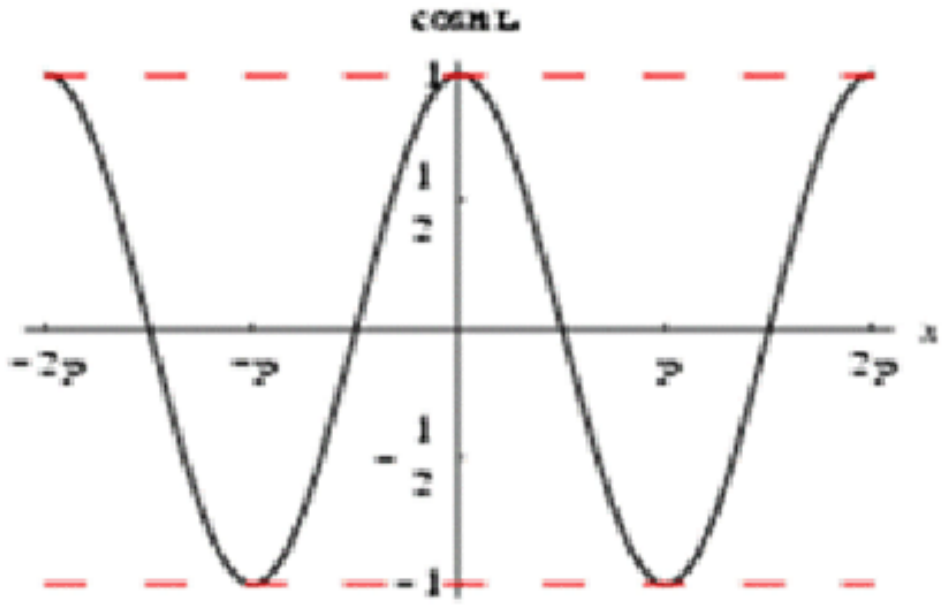
(3, 8, 4) x (2, 3, 1) = (3 x 2) + (8 x 3) + (4 x 1) = 34

Calculating Similarity in Vector Models

- Qualitatively or intuitively this means that documents are more likely to be about the same topics when they use the same terms in the same proportions
- When vectors are normalized (to length 1) compensate for different lengths, the inner product calculation is the cosine between the vectors

Cosines





"One of the basic trigonometric functions encountered in trigonometry. Let theta be an angle measured counterclockwise from the x-axis along the arc of the unit circle. Then cos(theta) is the horizontal coordinate of the arc endpoint. As a result of this definition, the cosine function is periodic with period 2pi."

Cosine of 0 degrees = 1; Cosine of 90 degrees = 0

Cosine Calculation – Normalized Vectors

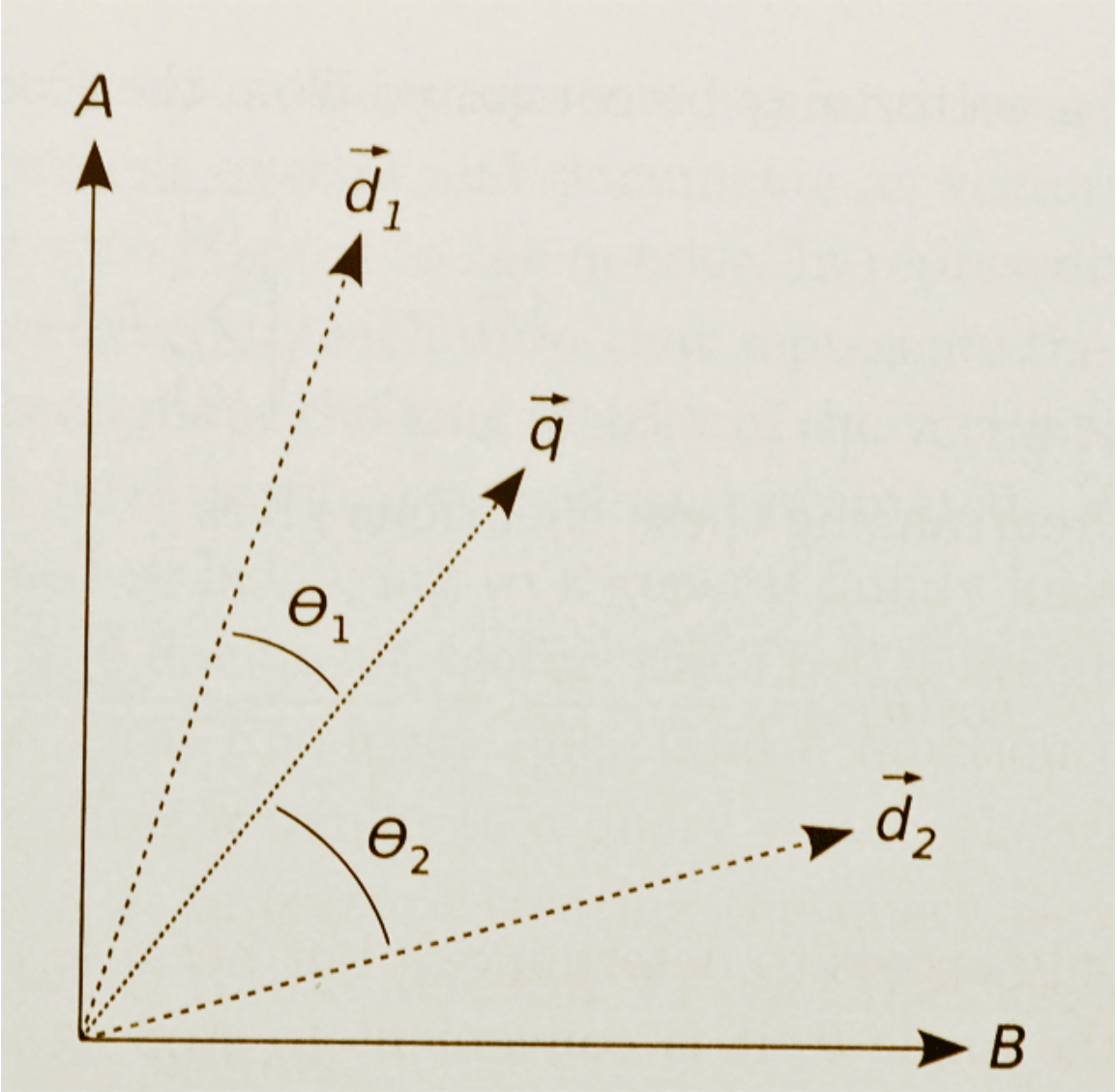
v1 = (3, 8, 4) v2 = (2, 3, 1)

(3, 8, 4) x (2, 3, 1) = (3 x 2) + (8 x 3) + (4 x 1) = 34

$$cos(\theta) = \frac{a \cdot b}{||a|| \cdot ||b||} = \frac{34}{\sqrt{89} \cdot \sqrt{14}} = \textbf{.992}$$

$\theta = arccos$ **(.992) = 15.59 degrees**

Similarity in Vector Models (Graphical Depiction)



Vector Model Retrieval and Ranking

- Vector models treat documents in a collection as "bags of words" so there is no representation of the order in which the terms occur in the document
- Not caring about word order lets us embody all the information about term occurrence in the term weights
- Likewise, vector queries are just "bags of words“

Vector Model Retrieval and Ranking

- So vector queries are fundamentally a form of "evidence accumulation" where the presence of more query terms in a document adds to its "score"
- This score is not an exact measure of relevance with respect to the query, but it is vastly better than the all or none Boolean model!

Vector Model: Advantages

- Index terms can be selected automatically
- Term weighting to improve retrieval performance
- Partial matching of queries and documents when no document contains all search terms
- Relevance ranking according to similarity

Vector Model: Limitations

- The calculations used by simple vector models are about the frequency of words and word forms (e.g., stemmed) in texts
- This means that they are measuring the "surface" usage of words as patterns of letters
- Polysemy leads to overestimates of similarity
- Synonymy leads to underestimates

Readings for Next Lecture

- Hearst, Marti – "Search User Interfaces" (Preface and Ch. 1)
- Hearst, Marti A. "'Natural'search user interfaces." Communications of the ACM 54, no. 11 (2011): 60-67. dl.acm.org/citation.cfm?id=2018414