# 23. Vector Models

**INFO 202 - 17 November 2008**

**Bob Glushko**

## Plan for Today's Class

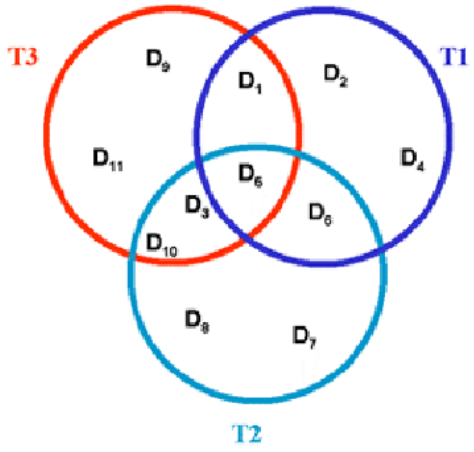
Relevance in the Boolean Model

The Vector Model

Term Weighting

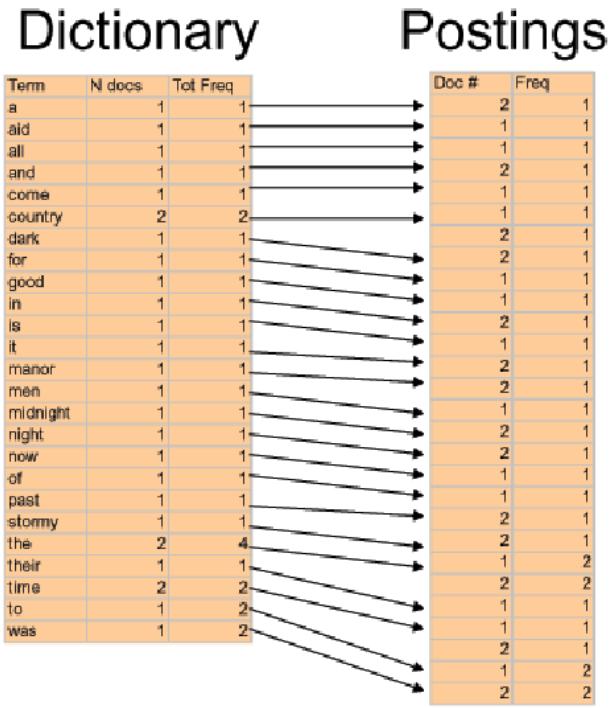
Similarity Calculation

# The Boolean Model




| *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 |

# Boolean Search with Inverted Indexes




**QUERY:**
**“time AND dark”**

**2 documents with “time” in dictionary have DOC#1, DOC#2 in posting file**

**1 document with “dark” in dictionary has DOC#2 in posting file**

**SOLUTION:**
**DOC#2 satisfies query**

## Relevance in the Boolean Model

In the Boolean model, documents and queries are represented as sets of index terms

So index terms are either present or absent in a document

How is the relevance of a document calculated?

On what basis are the retrieved documents ordered in a list presented to the searcher?

## Motivating Term Weighting from the Boolean Model

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

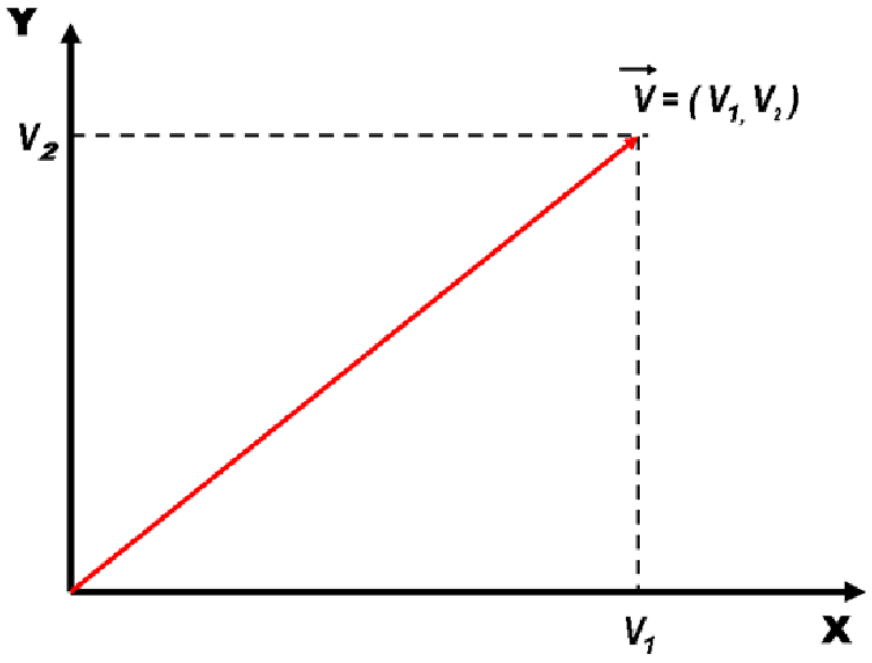
# Some Mathematical Foundations (and Review, I Hope)

Vectors

Summation Notation

Cosines

# Vectors [1]



# Vectors [2]

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

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)

# Summation Notation

$$s = \sum_{i=0}^{n-1} i$$

The Greek letter sigma is at the heart of summation notation, which specifies the sum of the terms in a sequence

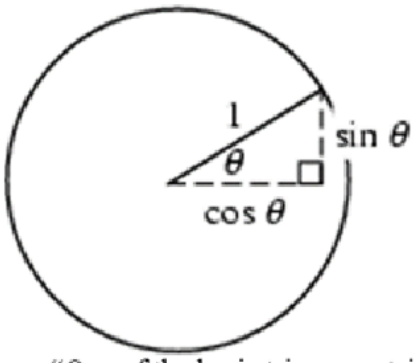
At the bottom and top of the sigma are integers that are the lower and upper limits of the term sequence

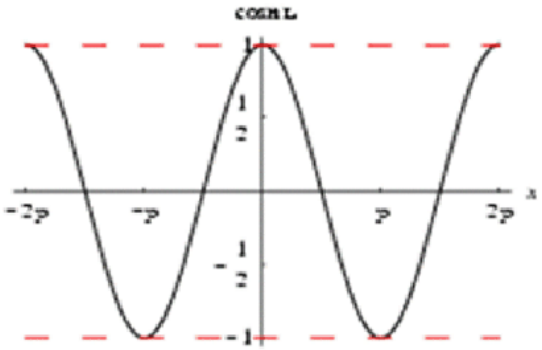
$n = 10$; $s = \sum_{i=0}^{n-1} i$

The index is $i$ and its boundaries are from 0 to $n$-1.

$s$ gets assigned the sum of all the integers from 0 to 9, inclusive or 0 + 1 + 2 + 3+4 + 5 + 6 + 7 + 8 + 9 = 45.

# 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."

# Overview of Vector Model

Documents and queries are represented as word or term vectors

Term weights can capture term counts within a document or the importance of the term in discriminating the document in the collection

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"

# An Important Note on Terminology

WARNING: A lot of IR literature uses *Frequency* to mean *Count*

- For example, *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

Unfortunately, this confused terminology is very entrenched and it would further confuse you if I tried to use more correct language, so I will conform to the incorrect usage

# Document x Term Matrix

We can create a matrix in which we 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 [1]

| 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 [2]

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

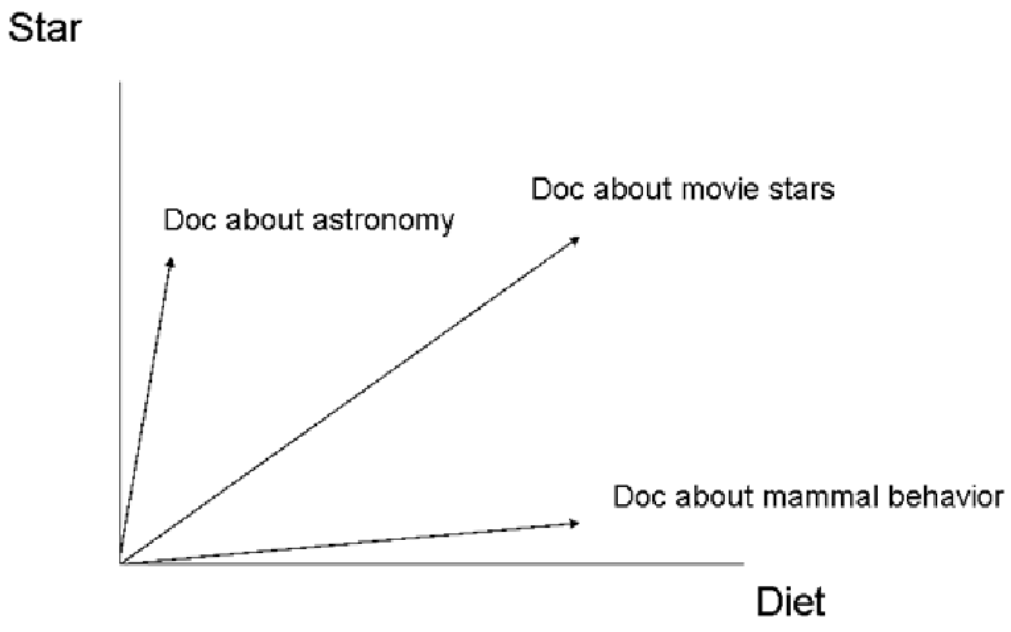
"Hollywood" occurs 7 times in text I
"Film" occurs 5 times in text I
"Diet" occurs 1 time in text I
"Fur" occurs 3 times in text I

# Word (or Term) Vectors

We can use this same matrix to think of the meaning of a word / terms as a vector whose coordinates measure how much the word 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 Example



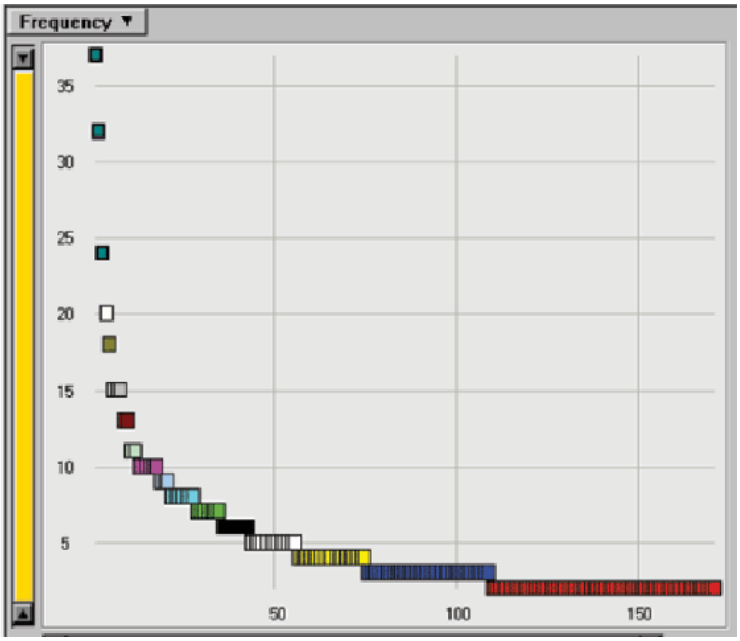
# A Small Text Collection (Stemmed)

| Rank | Freq | Term |
|---|---|---|
| 1 | 37 | system |
| 2 | 32 | knowledg |
| 3 | 24 | base |
| 4 | 20 | problem |
| 5 | 18 | abstract |
| 6 | 15 | model |
| 7 | 15 | languag |
| 8 | 15 | implem |
| 9 | 13 | reason |
| 10 | 13 | inform |
| 11 | 11 | expert |
| 12 | 11 | analysi |
| 13 | 10 | rule |
| 14 | 10 | program |
| 15 | 10 | oper |
| 16 | 10 | evalu |
| 17 | 10 | comput |
| 18 | 10 | case |
| 19 | 9 | gener |
| 20 | 9 | form |

| | | |
|---|---|---|
| 150 | 2 | enhanc |
| 151 | 2 | energi |
| 152 | 2 | emphasi |
| 153 | 2 | detect |
| 154 | 2 | desir |
| 155 | 2 | date |
| 156 | 2 | critic |
| 157 | 2 | content |
| 158 | 2 | consider |
| 159 | 2 | concern |
| 160 | 2 | compon |
| 161 | 2 | compar |
| 162 | 2 | commerci |
| 163 | 2 | clause |
| 164 | 2 | aspect |
| 165 | 2 | area |
| 166 | 2 | aim |
| 167 | 2 | affect |

# Stem Frequency Distribution for the Collection

| Rank | Freq | |
|---|---|---|
| 1 | 37 | system |
| 2 | 32 | knowledg |
| 3 | 24 | base |
| 4 | 20 | problem |
| 5 | 18 | abstract |
| 6 | 15 | model |
| 7 | 15 | languag |
| 8 | 15 | implem |
| 9 | 13 | reason |
| 10 | 13 | inform |
| 11 | 11 | expert |
| 12 | 11 | analysi |
| 13 | 10 | rule |
| 14 | 10 | program |
| 15 | 10 | oper |
| 16 | 10 | evalu |
| 17 | 10 | comput |
| 18 | 10 | case |
| 19 | 9 | gener |
| 20 | 9 | form |



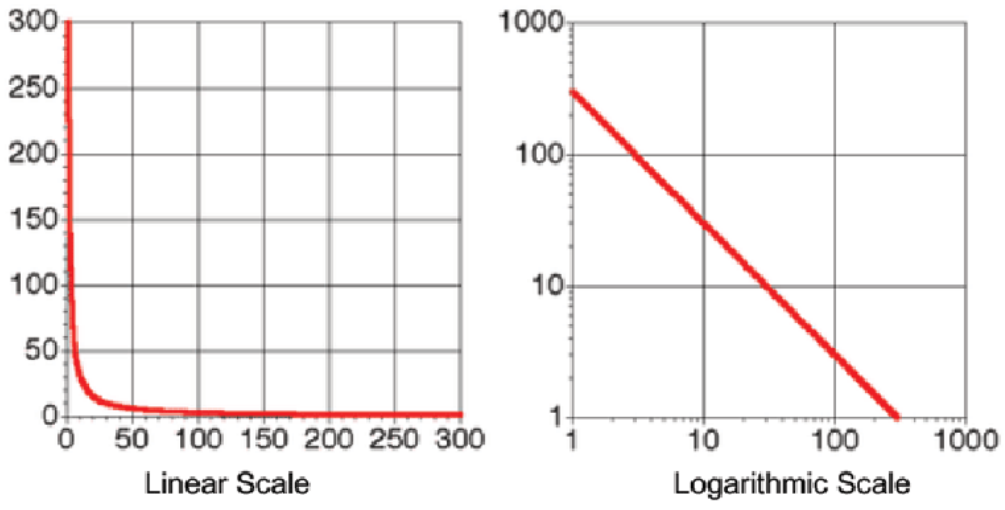
# The Zipf Distribution

We observe that:

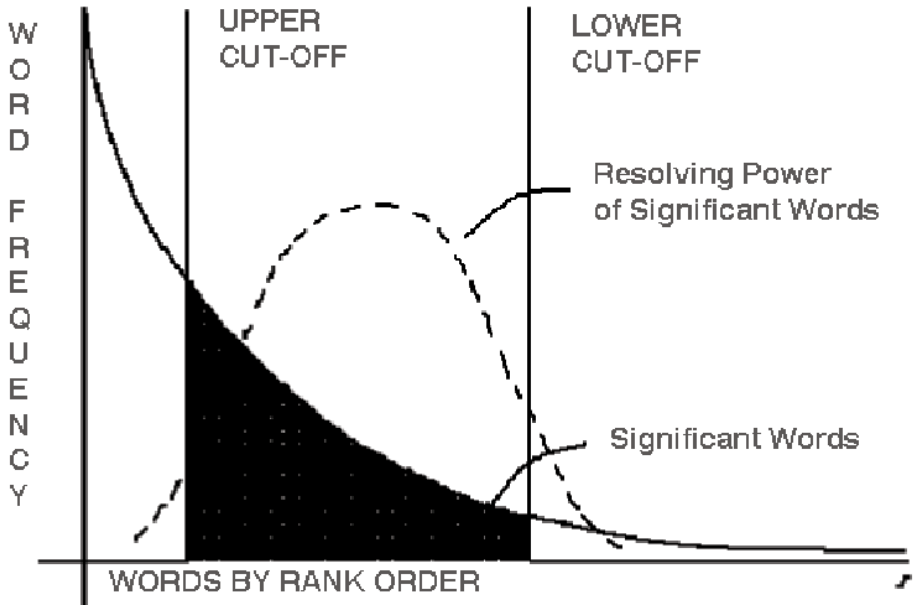
- A few items occur very frequently
- A medium number of elements have medium frequency
- Very many elements occur very infrequently (the "long tail")

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.

# Zipf Distribution - Linear vs Log Plots



# Word Frequency vs Discriminability / Resolving Power




# Same Idea, for Left-Brain Folks

Keywords, index terms, controlled vocabulary terms -- are not strictly properties of any single document. They reflect a relationship between an individual document and the set of documents it belongs to, from which it might be selected

The value of a potential keyword 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, with most of the occurrences in a small number of documents out of the collection

# Weighting Using Term Frequency

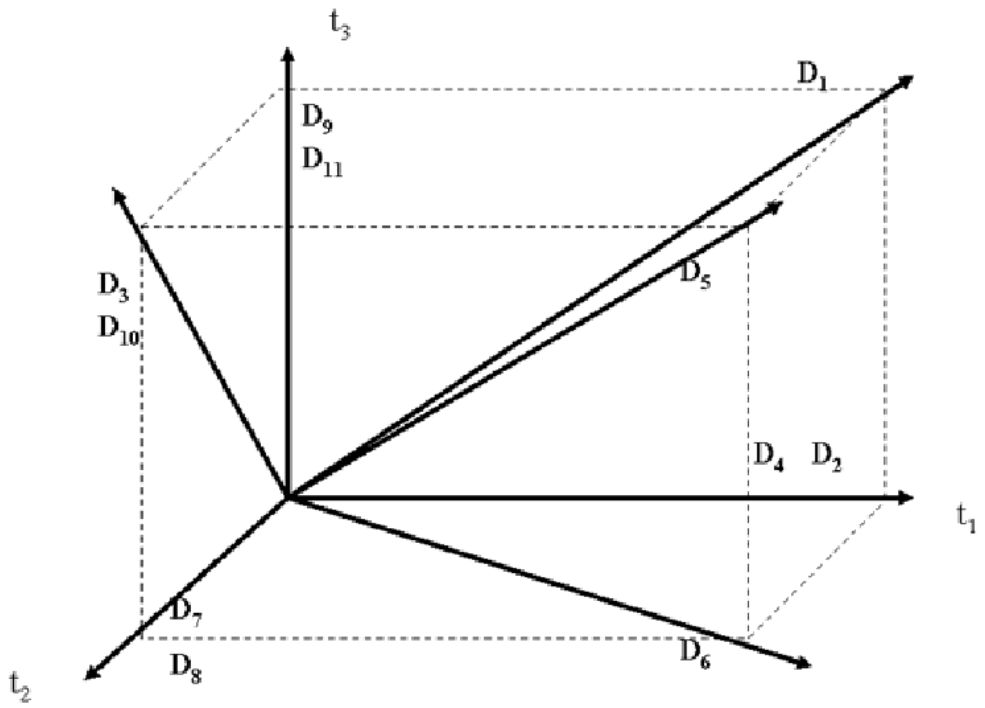
| docs | t1 | t2 | t3 |
|---|---|---|---|
| D1 | 2 | 0 | 3 |
| D2 | 1 | 0 | 0 |
| D3 | 0 | 4 | 7 |
| D4 | 3 | 0 | 0 |
| D5 | 1 | 6 | 3 |
| D6 | 3 | 5 | 0 |
| D7 | 0 | 8 | 0 |
| D8 | 0 | 10 | 0 |
| D9 | 0 | 0 | 1 |
| D10 | 0 | 3 | 5 |
| D11 | 4 | 0 | 1 |

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 in 3D



## Term Weighting -- Intuitions

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 they are by definition rare terms that most people will never use in queries

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

## 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 -- Calculation

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

$= \log (N / df_t)$

## "Inverse" Document Frequency -- 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 |

## Weighting Term Frequency with IDF (Simplified)

$tf_{td}$ = 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* )

## tf x idf Example Calculations

| | tf(1) | tf(2) | tf(3) | idf(1) log(N/n1) | idf(2) log(N/n2) | idf(3) log(N/n3) | w(1) | w(2) | w(3) |
|---|---|---|---|---|---|---|---|---|---|
| D1 | 2 | 0 | 3 | 0.176 | 0.273 | 0.331 | 0.352 | 0.000 | 0.993 |
| D2 | 1 | 0 | 0 | 0.176 | 0.273 | 0.331 | 0.176 | 0.000 | 0.000 |
| D3 | 0 | 4 | 7 | 0.176 | 0.273 | 0.331 | 0.000 | 1.092 | 2.317 |
| D4 | 3 | 0 | 0 | 0.176 | 0.273 | 0.331 | 0.528 | 0.000 | 0.000 |
| D5 | 1 | 6 | 0 | 0.176 | 0.273 | 0.331 | 0.176 | 1.638 | 0.000 |
| D6 | 3 | 5 | 0 | 0.176 | 0.273 | 0.331 | 0.528 | 1.365 | 0.000 |
| D7 | 0 | 8 | 0 | 0.176 | 0.273 | 0.331 | 0.000 | 2.184 | 0.000 |
| D8 | 0 | 10 | 0 | 0.176 | 0.273 | 0.331 | 0.000 | 2.730 | 0.000 |
| D9 | 0 | 0 | 1 | 0.176 | 0.273 | 0.331 | 0.000 | 0.000 | 0.331 |
| D10 | 0 | 3 | 5 | 0.176 | 0.273 | 0.331 | 0.000 | 0.819 | 1.655 |
| D11 | 4 | 0 | 1 | 0.176 | 0.273 | 0.331 | 0.704 | 0.000 | 0.331 |
| D12 | 1 | 0 | 3 | 0.176 | 0.273 | 0.331 | 0.176 | 0.000 | 0.993 |
| D13 | 5 | 1 | 0 | 0.176 | 0.273 | 0.331 | 0.880 | 0.273 | 0.000 |
| D14 | 9 | 0 | 0 | 0.176 | 0.273 | 0.331 | 1.585 | 0.000 | 0.000 |
| D15 | 3 | 1 | 2 | 0.176 | 0.273 | 0.331 | 0.528 | 0.273 | 0.662 |
| | 15 | 10 | 8 | 7 | | | | | |

## Normalized tf x idf

- Normalize the term weights (so longer vectors are not unfairly given more weight)
  - Force all values to fall within a certain range, usually between 0 and 1, inclusive

$$w_{ik} = \frac{tf_{ik} \log(N / \mathrm{df}_k)}{\sqrt{\sum_{k=1}^{t} (tf_{ik})^2 [\log(N / \mathrm{df}_k)]^2}}$$

## Normalized tf x idf Example Calculations

| | A | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|---|
| | tf(1)^2 | tf(2)^2 | tf(3)^2 | A x idf^2 | B x idf^2 | C x idf^2 | D+E+F | G^(1/2) |
| D1 | 4 | 0 | 9 | 0.124 | 0.000 | 0.986 | 1.1100 | 1.054 |
| D2 | 1 | 0 | 0 | 0.031 | 0.000 | 0.000 | 0.0310 | 0.176 |
| D3 | 0 | 16 | 49 | 0.000 | 1.192 | 5.368 | 6.5607 | 2.561 |
| D4 | 9 | 0 | 0 | 0.279 | 0.000 | 0.000 | 0.2791 | 0.528 |
| D5 | 1 | 36 | 0 | 0.031 | 2.683 | 0.000 | 2.7141 | 1.647 |
| D6 | 9 | 25 | 0 | 0.279 | 1.863 | 0.000 | 2.1423 | 1.464 |
| D7 | 0 | 64 | 0 | 0.000 | 4.770 | 0.000 | 4.7699 | 2.184 |
| D8 | 0 | 100 | 0 | 0.000 | 7.453 | 0.000 | 7.4530 | 2.730 |
| D9 | 0 | 0 | 1 | 0.000 | 0.000 | 0.110 | 0.1096 | 0.331 |
| D10 | 0 | 9 | 25 | 0.000 | 0.671 | 2.739 | 3.4097 | 1.847 |
| D11 | 16 | 0 | 1 | 0.496 | 0.000 | 0.110 | 0.6057 | 0.778 |
| D12 | 1 | 0 | 9 | 0.031 | 0.000 | 0.986 | 1.0170 | 1.008 |
| D13 | 25 | 1 | 0 | 0.775 | 0.075 | 0.000 | 0.8497 | 0.922 |
| D14 | 81 | 0 | 0 | 2.512 | 0.000 | 0.000 | 2.5117 | 1.585 |
| D15 | 9 | 1 | 4 | 0.279 | 0.075 | 0.438 | 0.7918 | 0.890 |

| | | | |
|---|---|---|---|
| idf: | 0.176 | 0.273 | 0.331 |
| idf^2 | 0.031 | 0.075 | 0.110 |

## Normalized tf x idf Example Calculations

| | norm w(1) | norm w(2) | norm w(3) |
|---|---|---|---|
| D1 | 0.3343 | 0.00 | 0.94 |
| D2 | 1 | 0.00 | 0.00 |
| D3 | 0 | 0.43 | 0.90 |
| D4 | 1 | 0.00 | 0.00 |
| D5 | 0.1069 | 0.99 | 0.00 |
| D6 | 0.3609 | 0.93 | 0.00 |
| D7 | 0 | 1.00 | 0.00 |
| D8 | 0 | 1.00 | 0.00 |
| D9 | 0 | 0.00 | 1.00 |
| D10 | 0 | 0.44 | 0.90 |
| D11 | 0.9051 | 0.00 | 0.43 |
| D12 | 0.1746 | 0.00 | 0.98 |
| D13 | 0.9551 | 0.30 | 0.00 |
| D14 | 1 | 0.00 | 0.00 |
| D15 | 0.5937 | 0.31 | 0.74 |

## Similarity in Vector Models

$$sim(D_i, D_j) = \sum_{k=1}^{t} w_{ik} * w_{jk}$$

- Similarity of two documents is calculated using normalized weights
- This is also called the cosine normalized inner product

# Cosine Similarity with Weighting Example Calculations

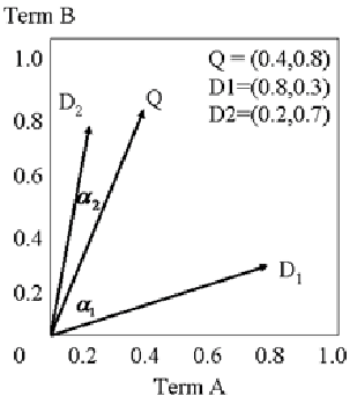
| | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| D1 | 1.00 | | | | | | | | | | | | | | |
| D2 | 0.33 | 1.00 | | | | | | | | | | | | | |
| D3 | 0.85 | 0.00 | 1.00 | | | | | | | | | | | | |
| D4 | 0.33 | 1.00 | 0.00 | 1.00 | | | | | | | | | | | |
| D5 | 0.04 | 0.11 | 0.42 | 0.11 | 1.00 | | | | | | | | | | |
| D6 | 0.12 | 0.36 | 0.40 | 0.36 | 0.97 | 1.00 | | | | | | | | | |
| D7 | 0.00 | 0.00 | 0.43 | 0.00 | 0.99 | 0.93 | 1.00 | | | | | | | | |
| D8 | 0.00 | 0.00 | 0.43 | 0.00 | 0.99 | 0.93 | 1.00 | 1.00 | | | | | | | |
| D9 | 0.94 | 0.00 | 0.90 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | | | | | | |
| D10 | 0.84 | 0.00 | 1.00 | 0.00 | 0.44 | 0.41 | 0.44 | 0.44 | 0.90 | 1.00 | | | | | |
| D11 | 0.70 | 0.91 | 0.38 | 0.91 | 0.10 | 0.33 | 0.00 | 0.00 | 0.43 | 0.38 | 1.00 | | | | |
| D12 | 0.99 | 0.17 | 0.89 | 0.17 | 0.02 | 0.06 | 0.00 | 0.00 | 0.98 | 0.88 | 0.58 | 1.00 | | | |
| D13 | 0.32 | 0.96 | 0.13 | 0.96 | 0.40 | 0.62 | 0.30 | 0.30 | 0.00 | 0.13 | 0.86 | 0.17 | 1.00 | | |
| D14 | 0.33 | 1.00 | 0.00 | 1.00 | 0.11 | 0.36 | 0.00 | 0.00 | 0.00 | 0.00 | 0.91 | 0.17 | 0.96 | 1.00 | |
| D15 | 0.90 | 0.59 | 0.80 | 0.59 | 0.37 | 0.50 | 0.31 | 0.31 | 0.74 | 0.80 | 0.85 | 0.84 | 0.66 | 0.59 | 1.00 |

# Similarity in Unnormalized Vectors

$$sim(Q, D_i) = \frac{\sum_{j=1}^{t} w_{qj} * w_{d_{ij}}}{\sqrt{\sum_{j=1}^{t} (w_{qj})^2 * \sum_{j=1}^{t} (w_{d_{ij}})^2}}$$

If the weights are not already normalized, we can combine the normalization and the similarity calculation using this equation

# Similarity in Unnormalized Vectors -- Example




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

$$sim(Q, D2) = \frac{(0.4 \cdot 0.2) + (0.8 \cdot 0.7)}{\sqrt{[(0.4)^2 + (0.8)^2] \cdot [(0.2)^2 + (0.7)^2]}}$$

$$= \frac{0.64}{\sqrt{0.42}} = 0.98$$

$$sim(Q, D_1) = \frac{.56}{\sqrt{0.58}} = 0.74$$

# 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"

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!

# Readings for 11/19

Susan Dumais, "Data-driven approaches to information access," Cognitive Science, 27(3), 491-524 (2003)

Clara Yu, John Cuadrado, Maciej Caglowski, & J. Scott Payne, "Patterns in Unstructured Data," 2002 (read from "Latent Semantic Indexing" through "Applications of LSI")