Lecture 3: Similarity

COMP90049
Knowledge Technologies

Lea Frermann and Karin Verspoor, CIS

Semester 2, 2019

Overview

Contact

Lea Frermann
Lecturer in Natural Language Processing (CIS)
lea.frermann@unimelb.edu.au

Please,

- Ask questions (!!)
- On-demand office hours (email me)
- Give feedback anytime (after lecture, email, ...)

Mid-semester Exam

When?	Friday, August 30 2019, **8.30am – 9:40 am**
Where?	Wilson Hall and Kwong Lee Dow
What?	50 minutes, with no reading time Closed-book, no materials needed All course content up to Friday, August 23rd (inclusive)
Weight?	10 marks

A sample exam (with solutions) is available in the course LMS → Assessment.

Recap

Knowledge Technologies

- processing **unstructured data** (text)
- **extracting** and **analyzing** information
- Part 1: Basics
- Part 2: Information Retrieval
- Part 3: Machine learning

Last Week

- Regular expressions
- you know what you want to find **and** where to look

Recap

Knowledge Technologies

- processing **unstructured data** (text)
- **extracting** and **analyzing** information
- Part 1: Basics
- Part 2: Information Retrieval
- Part 3: Machine learning

Last Week

- Regular expressions
- you know what you want to find **and** where to look
- But what if you don't? How do you judge what is **relevant enough**?

Similarity – So what?!

- humans can’t help but group items into classes (categories)
- categories drive how we perceive and make decisions

https://commons.wikimedia.org/wiki/File:USDA_-_Basic_7_Food_Groups.jpg

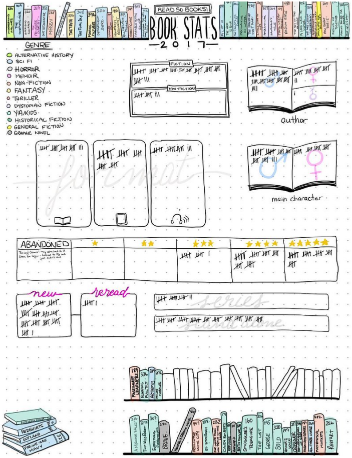
Similarity – So what?!

- humans can't help but group items into classes (categories)
- categories drive how we perceive and make decisions

https://www.flickr.com/photos/mikecogh/7561440544

Similarity – So what?!

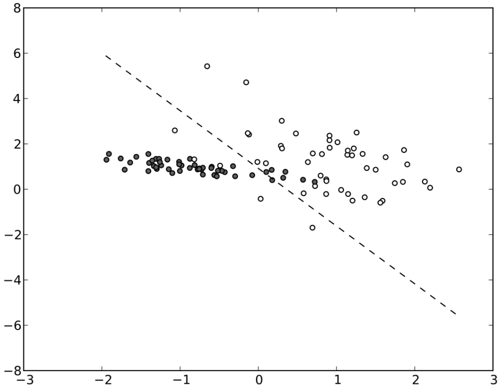
- humans can't help but group items into classes (categories)
- categories drive how we perceive and make decisions



https://www.flickr.com/photos/dolmansaxlil/23491147328

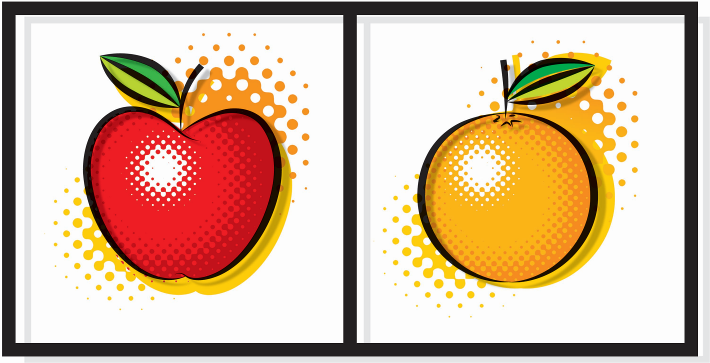
Similarity – So what?!

- humans can't help but group items into classes (categories)
- categories drive how we perceive and make decisions
- ...and we want to **develop knowledge technologies** to help us with this

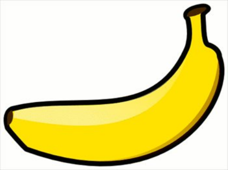


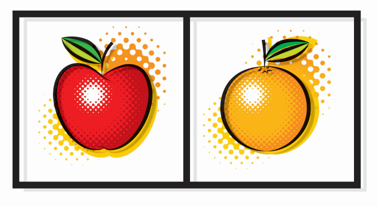
https://commons.wikimedia.org/wiki/File:Linear-svm-scatterplot.svg

Compare and Contrast

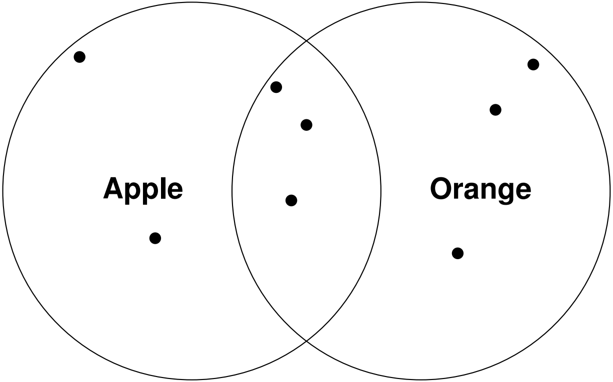


Compare and Contrast

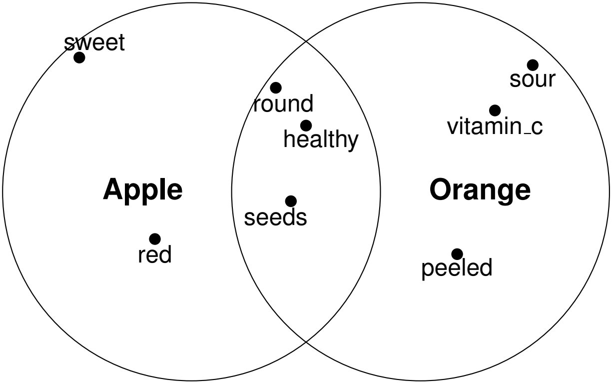




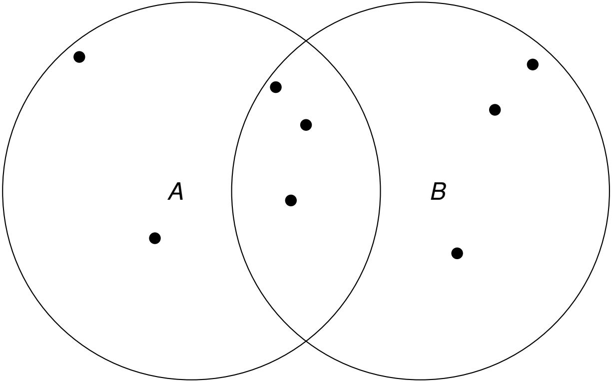
Venn Diagram



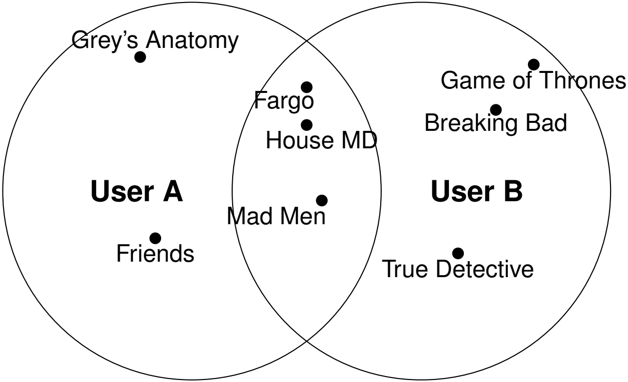
Venn Diagram



Venn Diagram



Venn Diagram



How similar is User A to User B?

Similarity as Set intersection

Many similarity assessments can be framed as set intersection.

- Amazon: Book purchases
- Netflix: Movies that you have watched

Refinements

- Rating sets (stars)
  - thresholding using ratings
  - different subsets for different ratings
- Categories of items
  - generalisation
  - book or movie genres

Comparing Documents

How should we compare documents to assess their similarity?

- String-level similarity (e.g., edit distance)
- Sets of common substrings (sentences, phrases, words, n-grams)
- “bag of words”
- Meaning (whatever that is?)

Comparing Documents

How should we compare documents to assess their similarity?

- String-level similarity (e.g., edit distance)
- Sets of common substrings (sentences, phrases, words, n-grams)
- "bag of words"
- Meaning (whatever that is?)

How similar are these sentences?

a. Mary is quicker than John. $\nearrow$ b. Mary is slower than John.

a. Mary is quicker than John. $\searrow$ c. John is quicker than Mary.

Word Vectors

How similar are these sentences?

a. Mary is quicker than John.

→ b. Mary is slower than John.

→ c. John is quicker than Mary.

Sentence	"Mary"	"John"	"quicker"	"slower"
a.	1	1	1	0
b.	1	1	0	1
c.	1	1	1	0

Word Vectors

How similar are these sentences?

a. Mary is quicker than John. → b. Mary is slower than John.

a. Mary is quicker than John. → c. John is quicker than Mary.

Features

Sentence	“Mary”	“John”	“quicker”	“slower”
a.	1	1	1	0
b.	1	1	0	1
c.	1	1	1	0

Values

Feature vectors

A **feature** or *attribute* is any distinct aspect, quality, or characteristic of that object

Features may be:

Feature vectors

A **feature** or *attribute* is any distinct aspect, quality, or characteristic of that object

Features may be:

- symbolic/categorical/discrete (e.g. colour, gender)

Feature vectors

A **feature** or *attribute* is any distinct aspect, quality, or characteristic of that object

Features may be:

- symbolic/categorical/discrete (e.g. colour, gender)
- ordinal (e.g. cool $<$ mild $<$ hot [temperature])

Feature vectors

A **feature** or *attribute* is any distinct aspect, quality, or characteristic of that object

Features may be:

- symbolic/categorical/discrete (e.g. colour, gender)
- ordinal (e.g. cool $<$ mild $<$ hot [temperature])
- numeric/continuous (e.g., height, age)

Feature vectors

A **feature** or *attribute* is any distinct aspect, quality, or characteristic of that object

Features may be:

- symbolic/categorical/discrete (e.g. colour, gender)
- ordinal (e.g. cool $<$ mild $<$ hot [temperature])
- numeric/continuous (e.g., height, age)
- binary (e.g., has_feathters, is_round)

Feature vectors

A **feature** or *attribute* is any distinct aspect, quality, or characteristic of that object

Features may be:

- symbolic/categorical/discrete (e.g. colour, gender)
- ordinal (e.g. cool $<$ mild $<$ hot [temperature])
- numeric/continuous (e.g., height, age)
- binary (e.g., has_feathters, is_round)

Feature vectors

A **feature** or *attribute* is any distinct aspect, quality, or characteristic of that object

Features may be:

- symbolic/categorical/discrete (e.g. colour, gender)
- ordinal (e.g. cool $<$ mild $<$ hot [temperature])
- numeric/continuous (e.g., height, age)
- binary (e.g., has_feathters, is_round)

A **feature vector** is an n-dimensional vector of *features* that represent some object.

Feature vectors

A **feature** or *attribute* is any distinct aspect, quality, or characteristic of that object

Features may be:

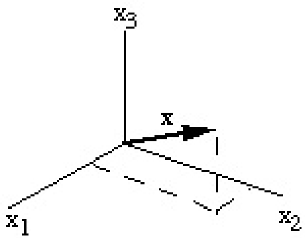
- symbolic/categorical/discrete (e.g. colour, gender)
- ordinal (e.g. cool $<$ mild $<$ hot [temperature])
- numeric/continuous (e.g., height, age)
- binary (e.g., has_feathters, is_round)

A **feature vector** is an n-dimensional vector of *features* that represent some object.

A vector **locates** an object (document, person, . . .) as a point in *n*-dimensional **space**. The **angle** of the vector in that space is determined by the relative **weight** of each term.

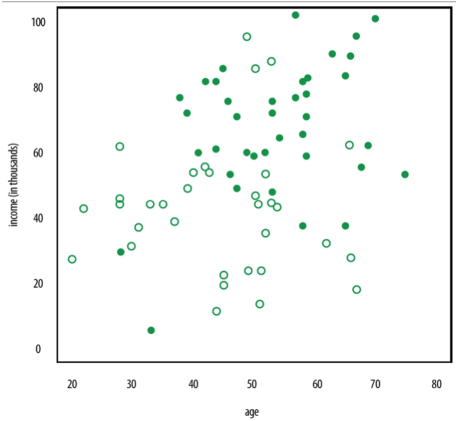
Feature vectors and vector space

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_d \end{bmatrix}$$



Credit as a function of age and income

age	income	credit
33	8	low
58	42	low
49	79	low
49	17	low
58	26	high
44	71	high
...		



Activity – features for spam prediction

In pairs, think of (at least) one **conituous**, one **categorical** and one **binary** feature predictive for spam vs non-spam e-mails.

title	re [8] : dear friend
body	ORDER CONFIRMATION!!! your order should be shipped by january, via fedex. your federal express tracking number is 45954036. thank you for registering. your userid is: 56075519 learn to make a fortune with ebay! complete turnkey system software - videos - turorials clk here for information clilings.
title	vacation payback
body	I just received a call from payroll - they miscalculated your vacation payback. Rather than the $1,113.66, you actually only owe $957.04 (they forgot to take away your or state taxes in the first calculation). Can you please write another check and we will void the first one? Sorry for the inconvenience. Please let me know if you have any questions. Amy

Examples from Enron4 spam data: http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html

Activity – features for spam prediction

In pairs, think of (at least) one **conituous**, one **categorical** and one **binary** feature predictive for spam vs non-spam e-mails.

Continuous features

- average word length (body)
- average word length (title)
- time of sending (continuous)

Categorical features

- title length
- message length
- topic
- time of sending (binned)

Binary features

- word occurrence
- non-empty title
- contains capitalized words
- contains URL
- contains repeated punctuation

Vector space model for document retrieval

- proposed in 1962, one of the earliest document **retrieval** models

Vector space model for document retrieval

- proposed in 1962, one of the earliest document **retrieval** models

- *n* distinct indexed terms (aka. terms of interest) in the collection
- term importance weight $w_{d,t}$ for each document d
  e.g., count, binary, importance indicator (tf-idf, next)

Vector space model for document retrieval

- proposed in 1962, one of the earliest document **retrieval** models

- n distinct indexed terms (aka. terms of interest) in the collection
- term importance weight $w_{d,t}$ for each document d
  e.g., count, binary, importance indicator (tf-idf, next)

- documents d_i are represented as a vector

$$d_1 = \langle w_{d_1,1}, w_{d_1,2}, \ldots, w_{d_1,t}, \ldots, w_{d_1,n} \rangle$$
$$d_2 = \langle w_{d_2,1}, w_{d_2,2}, \ldots, w_{d_2,t}, \ldots, w_{d_2,n} \rangle$$

Vector space model for document retrieval

- proposed in 1962, one of the earliest document **retrieval** models

- n distinct indexed terms (aka. terms of interest) in the collection
- term importance weight $w_{d,t}$ for each document d
  e.g., count, binary, importance indicator (tf-idf, next)

- documents d_i are represented as a vector

e.g., d_1 could be a query and d_2 a website

$$d_1 = \langle w_{d_1,1}, w_{d_1,2}, \ldots, w_{d_1,t}, \ldots, w_{d_1,n} \rangle$$
$$d_2 = \langle w_{d_2,1}, w_{d_2,2}, \ldots, w_{d_2,t}, \ldots, w_{d_2,n} \rangle$$

Vector space model for document retrieval

- proposed in 1962, one of the earliest document **retrieval** models

- n distinct indexed terms (aka. terms of interest) in the collection
- term importance weight $w_{d,t}$ for each document d
  e.g., count, binary, importance indicator (tf-idf, next)

- documents d_i are represented as a vector

e.g., d_1 could be a query and d_2 a website

$$d_1 = \langle w_{d_1,1}, w_{d_1,2}, \ldots, w_{d_1,t}, \ldots, w_{d_1,n} \rangle$$
$$d_2 = \langle w_{d_2,1}, w_{d_2,2}, \ldots, w_{d_2,t}, \ldots, w_{d_2,n} \rangle$$

- if the weights for d_1 and d_2 are similar, then they likely to be **similar in topic**
- the content of indivicual document d is limited and focussed, so that most $w_{d,t}$ values will be zero (**sparsity**).

Term Weighting

Document 1	Document 2	Document 3
Yesterday, Mary went to the store and bought a bunch of apples.	Yesterday, Peter talked to the agents and filed a bunch of complaints.	Agents typically ignore and delete the complaints.

- which document is most relevant to **Document 2**?
- which words are **informative**?
- what distributional properties should these words have?

Term Weighting

Document 1	Document 2	Document 3
Yesterday, Mary went **to the** store **and** bought **a bunch of** apples.	**Yesterday**, Peter talked **to the** agents **and** filed **a bunch of** complaints.	Agents typically ignore and delete the complaints.

- which document is most relevant to **Document 2**?
- which words are **informative**?
- what distributional properties should these words have?

Term Weighting

Document 1	Document 2	Document 3
Yesterday, Mary went to the store and bought a bunch of apples.	Yesterday, Peter talked to **the agents and** filed a bunch of **complaints**.	**Agents** typically ignore **and** delete **the complaints**.

- which document is most relevant to **Document 2**?
- which words are **informative**?
- what distributional properties should these words have?

Term Frequency-Inverse Document Frequency

We will formalize **two basic intuitions**: term weights are:

1. terms that occur frequently in a given document have high utility:

$$w_{d,t} \propto f_{d,t} \xleftarrow[\textbf{term t in doc d}]{f_{d,t}\textbf{: frequency of}}$$

Term Frequency-Inverse Document Frequency

We will formalize **two basic intuitions**: term weights are:

1. terms that occur frequently in a given document have high utility:

$$w_{d,t} \propto f_{d,t}$$

$f_{d,t}$: frequency of term t in doc d

2. terms that occur in a wide variety of documents have low utility:

$$w_t \propto \frac{1}{f_t}$$

f_t: # documents containing t

Term Frequency-Inverse Document Frequency

We will formalize **two basic intuitions**: term weights are:

1. terms that occur frequently in a given document have high utility:

$$w_{d,t} \propto f_{d,t}$$

$f_{d,t}$: **frequency of term t in doc d**

2. terms that occur in a wide variety of documents have low utility:

$$w_t \propto \frac{1}{f_t}$$

f_t: **# documents containing t**

Models which weigh up these two are referred to as **TF-IDF** (term frequency–inverse document frequency) models

The "classic" TF-IDF formulation is:

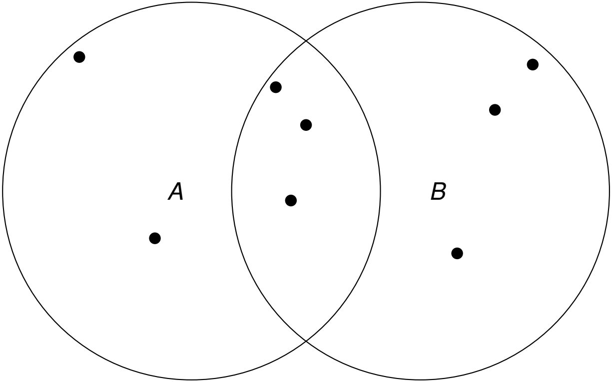
$$w_{d,t} = f_{d,t} \times \log \frac{N}{f_t}$$

N: **# documets in collection**

Measuring Similarity

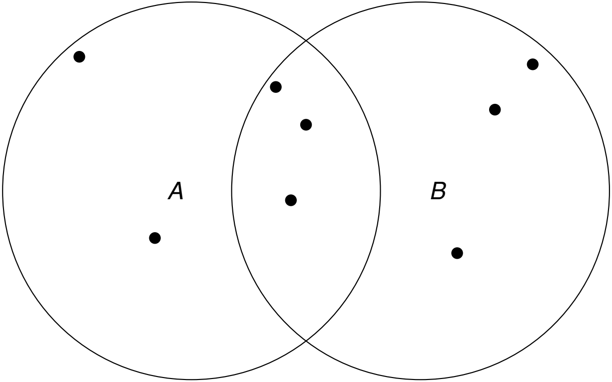
- ✓ We have discussed similarity at an intuitive level.
- ✓ We have formalized feature (term) weighting.
- ✗ How do we measure similarity quantitatively?

Jaccard Similarity



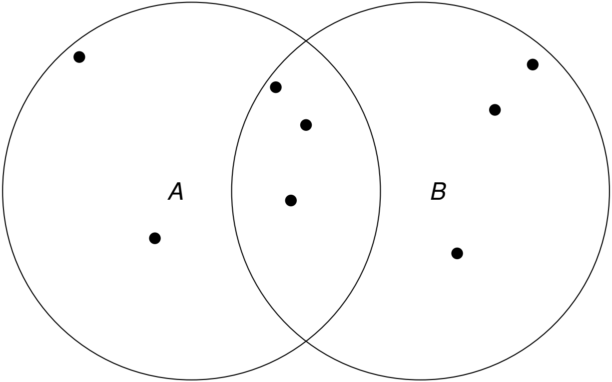
$$sim(A, B) = \frac{|A \cap B|}{|A \cup B|} = ?$$

Jaccard Similarity



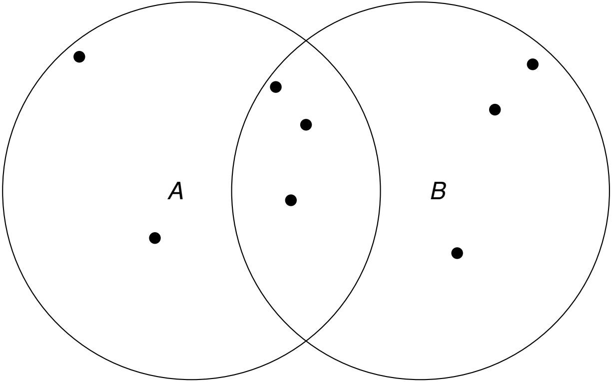
$$sim(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{3}{8}$$

Dice Similarity



$$sim(A, B) = \frac{2|A \cap B|}{|A| + |B|} = ?$$

Dice Similarity



$$sim(A, B) = \frac{2|A \cap B|}{|A| + |B|} = \frac{2 * 3}{5 + 6} = \frac{6}{11}$$

Similarity vs Distance

What is the relationship between similarity and distance?

Distance measures

A distance measure on a space is a function that takes two points in a space as arguments.

1. No negative distances.

$$d(x, y) \geq 0$$

2. Distances are positive, except for the distance from a point to itself.

$$d(x, y) = 0 \text{ if and only if } x = y$$

3. Distance is symmetric.

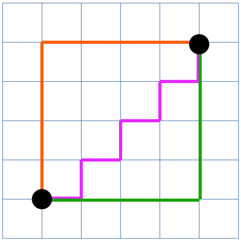
$$d(x, y) = d(y, x)$$

4. The *triangle inequality* typically holds.
(Distance measures the length of the *shortest path* between two points.)

$$d(x, y) \leq d(x, z) + d(z, y)$$

Manhattan Distance (aka L1 Distance)

- Given two items A and B, as their feature vectors $\vec{a}$ and $\vec{b}$
- Absolute differences of their cartesian coordinates
- Travel coordinate by coordinate. No short cuts!

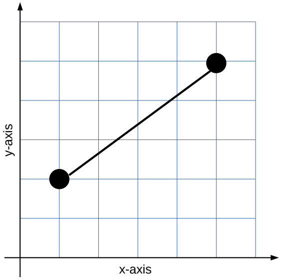


In n-dimensional space:

$$d(A, B) = \sum_{i=1}^{n} |a_i - b_i|$$

Euclidean Distance (aka L2 Distance)

- Given two items A and B, as their feature vectors $\vec{a}$ and $\vec{b}$
- Straight line between the two points

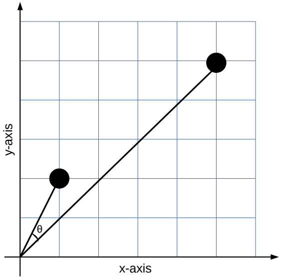


In n-dimensional space:

$$d(A,B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$

Cosine Distance

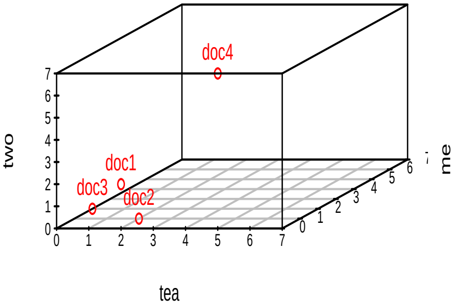
- Given two items A and B, as their feature vectors $\vec{a}$ and $\vec{b}$
- Similarity as the cosine of the angle θ between the two vectors



$$sim(A,B) = \frac{\vec{a}\cdot\vec{b}}{|\vec{a}||\vec{b}|} = \frac{\sum_i a_i b_i}{\sqrt{\sum_i a_i^2}\sqrt{\sum_i b_i^2}}$$

“Long” documents & Euclidean distance

	tea	me	two
doc1	2	0	2
doc2	2	1	0
doc3	0	2	0
doc4	5	0	7



- Doc4, like Doc1, is all about “tea” and “two”.
- But because it is longer, it is in a space by itself.

Probabilistic measures

Relative entropy:

$$D(x \mid\mid y) = \sum_i x_i(\log_2 x_i - \log_2 y_i)$$

or alternatively *skew divergence*:

$$s_\alpha(x, y) = D(x \mid\mid \alpha y + (1-\alpha)x)$$

or *Jensen-Shannon divergence*:

$$JSD(x \mid\mid y) = \frac{1}{2}D(x \mid\mid m) + \frac{1}{2}D(y \mid\mid m)$$

where $m = \frac{1}{2}(x+y)$

NB: Probability will be reviewed next lecture!

Summary

- How can we represent a set of objects?
- What are some methods for measuring similarity between objects?

Reading

- On distance measures:

Chapter 3, especially Section 3.5
Mining of Massive Datasets
`http://infolab.stanford.edu/~ullman/mmds.html`

- On document representation:

Chapter 6
Information Retrieval, Manning *et al.*
`http://nlp.stanford.edu/IR-book/html/htmledition/scoring-term-weighting-and-the-vector-space-model-1.html`