Learning Analytics

Dr. Bowen Hui
Computer Science
University of British Columbia Okanagan

When There Are Too Many Choices

- Browsing can be time consuming
  - May miss out relevant choices
  - Only view first few items
- Solution: Recommendation systems

- Applications in education:
  - Academic choices (e.g. electives)
  - Learning activities
  - Learning resources
  - Etc.

Learner Profiles and Personalization

- Idea: the more I know about a student, the more I can adapt my lesson to the student

Learner Profiles and Personalization

- Idea: the more I know about a student, the more I can adapt my lesson to the student

- Key questions:
  - How and what to learn about a student?
  - Which parts of the lesson can be adapted?
  - How do each adaptation affect different types of students?

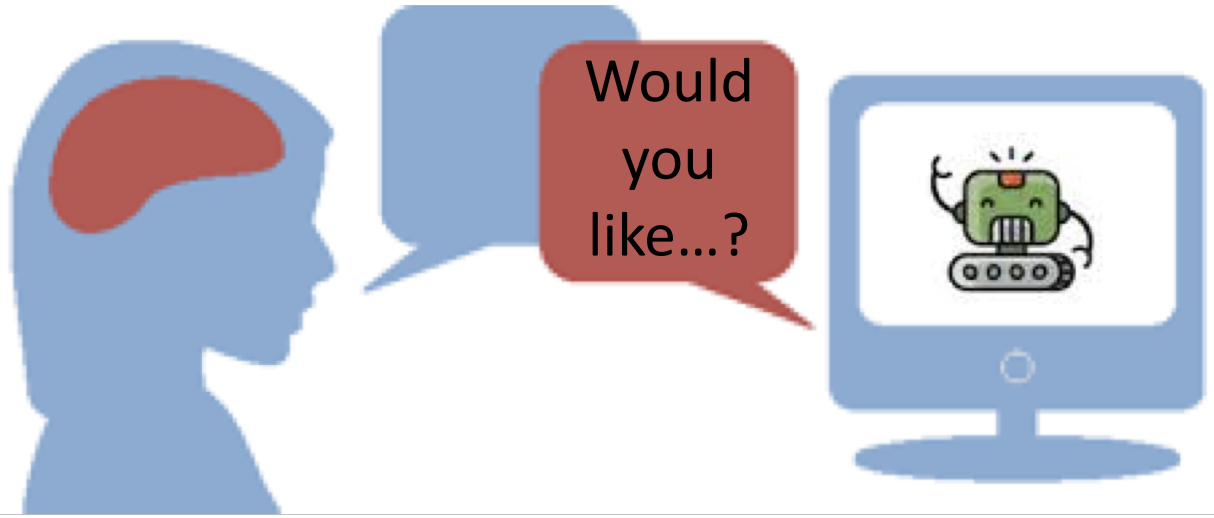
- Open research questions!

Viewing System Actions as Recommendations

- Example system actions
  - Hiding unused information
  - Moving objects to better locations
  - Auto-completions

Viewing System Actions as Recommendations

- Example system actions
  - Hiding unused information
  - Moving objects to better locations
  - Auto-completions



Two Major Types of Recommendation Systems

- Content based approach
  - Using content to understand what user likes

- Collaborative approach
  - Using other users to understand a given user likes
  - Also called collaborative filtering

How do these apply to learning and personalization?

Content Based Recommendations

- What content interests the user?
  - Webpage browsing history
  - Documents previously read
  - Books/movies/etc. the user liked

- Focus on implicit feedback and text content

- Task: Given a history of browsed documents and a new document, will the user like this new document?

Underlying Assumption

- Task: Given a history of browsed documents and a new document, will the user like this new document?

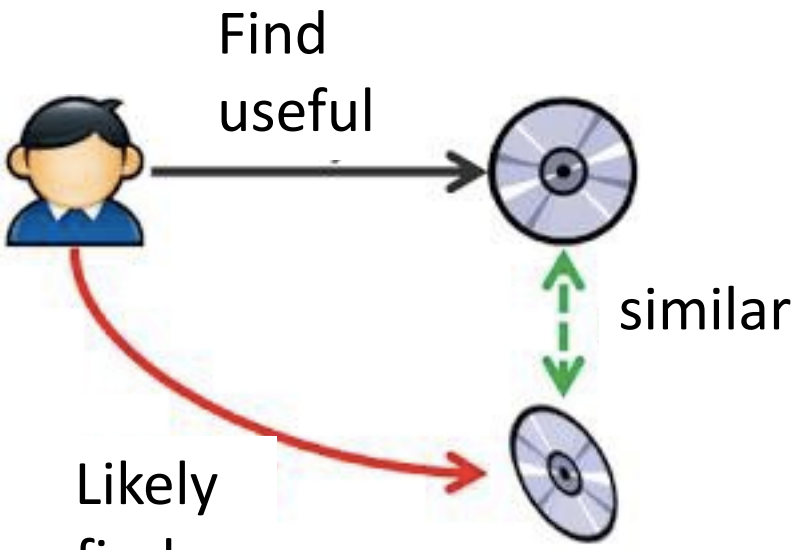


Image taken from towardsdatascience.com

Risk: recommendations have no diversity

How to Represent Documents?

Image taken from flylib.com

Vector Space Model!

Data Representation

- Ignores word order, syntax, semantics, higher level language context

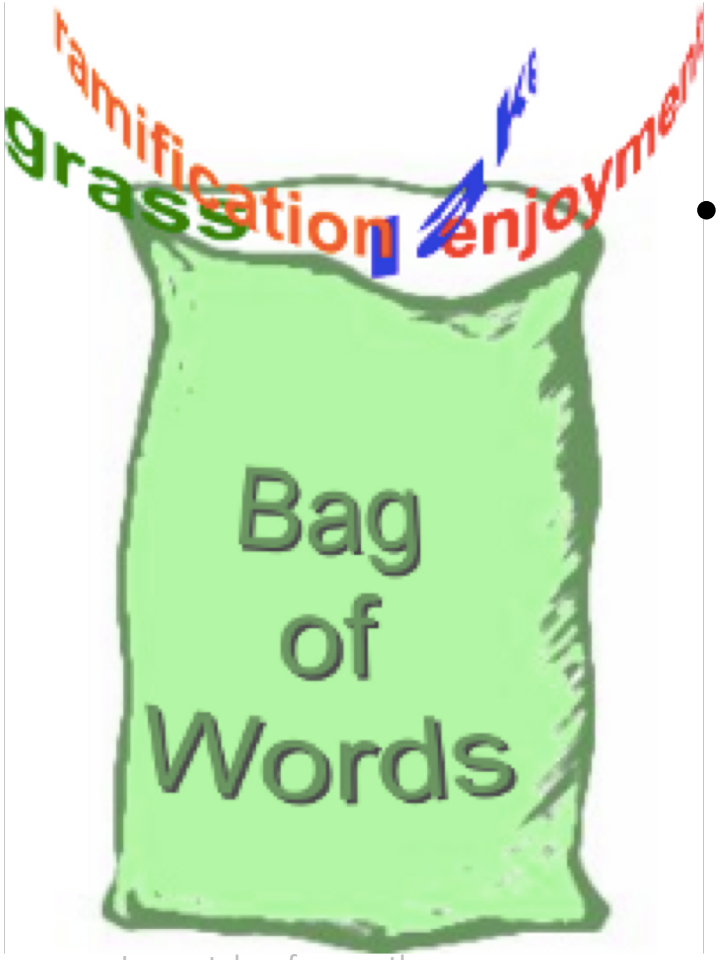
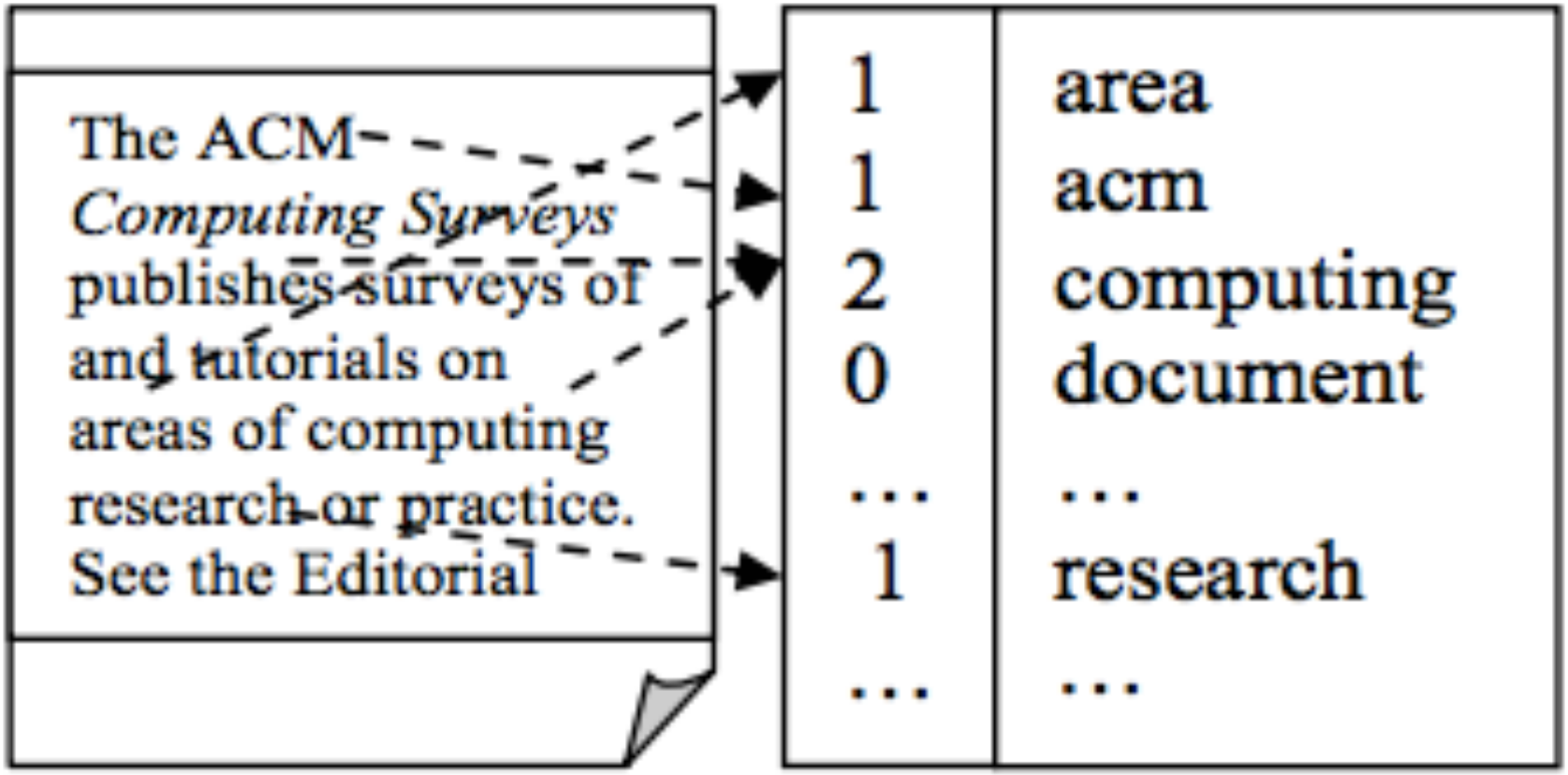


Image taken from python-course.eu

- Represent document as word vector

"John is quicker than Mary" treated same as "Mary is quicker than John"

Building a Document Vector



Reference: User Profiling for the Web (Grcar et al. 2006)

Space savings!

Need for Preprocessing

- Try creating a document vector for:

The dog jumped over the cat that was sitting on a log next to the swamp by the park yesterday. That was the highest jump the dog has ever jumped!

- Can we approximate overall meaning without keeping all the content?

Selection of Words

- Remove stop-words that do not contribute to document uniqueness
  - E.g., “a”, “the”, “with”
  - Stoplists exist for English
- Preprocess words to arrive at their stems (via a stemming process)
  - E.g., “singing” and “sings” both become “sing”
  - Look up the Porter stemmer algorithm
- Rule of thumb: Keep top 10% of most frequently occurring words to reduce complexity

Term Frequency, tf

- Raw term frequency doesn't work well
  - A doc with 10 occurrences of t may be more relevant than another doc with 1 occurrence of t
  - But not necessarily 10 times more relevant
  - Need: tf to increase proportionally w.r.t. relevance

- Given term t and document d:

$$tf(t,d) = \frac{\text{frequency of t in d}}{\text{total number of terms in d}}$$

Why do we divide by the number of terms in d?

Term Frequency, tf

- Raw term frequency doesn't work well
  - A doc with 10 occurrences of t may be more relevant than another doc with 1 occurrence of t
  - But not necessarily 10 times more relevant
  - Need: tf to increase proportionally w.r.t. relevance

- Given term t and document d:

$$tf(t,d) = \frac{\text{frequency of t in d}}{\text{total number of terms in d}}$$

  - Normalized so to not bias frequently occurring words in long documents

Frequency and Relevance to Meaning

- Most frequent word may not be the most relevant word to a document
- Less frequently occurring words may serve to better identify document relevance
- Need to consider how rare a word is in all other documents

Inverse Document Frequency, idf

- Given term t and set of documents D:

$$\text{idf}(t,D) = \log \frac{|D|}{\text{df}(t, D)}$$

Log is used to dampen the effect of idf

where df(t,D) is the document frequency

- Document frequency defined as the number of documents that contain t
  - Measures informativeness of t

idf example

- Suppose |D| = 1 million

Term	df	idf
calpurnia	1	
animal	100	
sunday	1,000	
fly	10,000	
under	100,000	
the	1,000,000	

What are the idf values?

- Every term has one idf value in a collection

idf example

- Suppose |D| = 1 million

Term	df	idf
calpurnia	1	6
animal	100	4
sunday	1,000	3
fly	10,000	2
under	100,000	1
the	1,000,000	0

Weigh down common terms but scale up rare ones

- Every term has one idf value in a collection

Overall Definition of tfidf

- Given document set D, a document d, and a term t:

$$\text{tf-idf}(t,d,D) = \text{tf}(t,d) \times \text{idf}(t,D)$$

- Increases with
  - The number of occurrences within a document
  - The rarity of the term in the document collection

- Best known weighting scheme in information retrieval

Document Representation

- Each document can be represented as a vector
  $d = (tfidf_1, tfidf_2, tfidf_3, \ldots)$

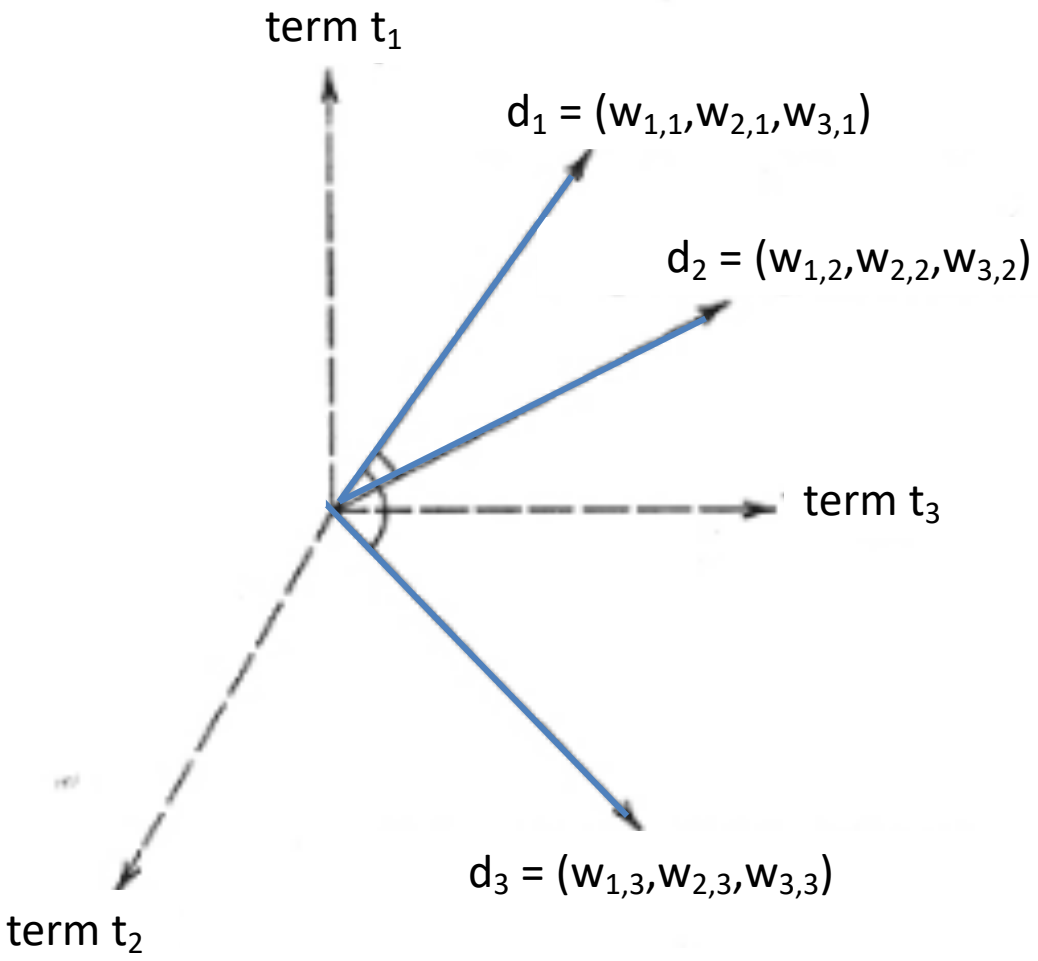
- More generally:
  $d_j = (w_{1,j}, w_{2,j}, w_{3,j}, \ldots)$
  - where $w_{i,j}$ is the tfidf weight for term i and document j in the collection of documents

Representing a Document Collection

- As a matrix:

	$\mathbf{t_1}$	$\mathbf{t_2}$	…	$\mathbf{t_i}$
d_1	$w_{1,1}$	$w_{2,1}$	…	$w_{i,1}$
d_2	$w_{1,2}$	$w_{2,2}$	…	$w_{i,2}$
…	…	…		…
d_j	$w_{1,j}$	$w_{2,j}$	…	$w_{i,j}$

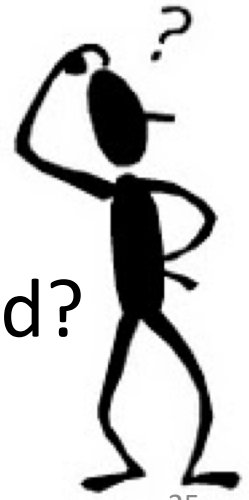
Document Space



Original Image taken from www.cs.uni.edu

So Far…

- We gathered a set of documents that the user has browsed through recently
- Each document is represented as a weighted vector
- There is a new document we don't know anything about
- Original task:
  Will the user like a new document at hand?



Measuring Document Similarity

- Need to compute degree of similarity between two vectors
  - Not just if two vectors are similar or not
  - But how similar they are

- How should we define sim(d_1, d_2) = ?

Inner Product

- Suppose sim(d_1, d_2) = $d_1 \bullet d_2$
- Recall definition:
  $d_1 \bullet d_2 = \sum_{i=1}^{t} d_{i1}\, di_2$

- Simple (binary) example:

	Mary	run	dog	jump	house	lives	today
d_1	1	1	1	0	1	1	0
d_2	1	0	1	0	0	1	1

sim(d_1, d_2) = 3

Properties of Inner Product

- Two identical vectors:
  - Dot product = their squared magnitude
- Two orthogonal vectors (i.e., perpendicular):
  - Dot product = 0

- Properties of inner product:
  - Unbounded
  - Favours long documents with large number of unique terms
  - Measures how many terms matched but not how many terms did not match

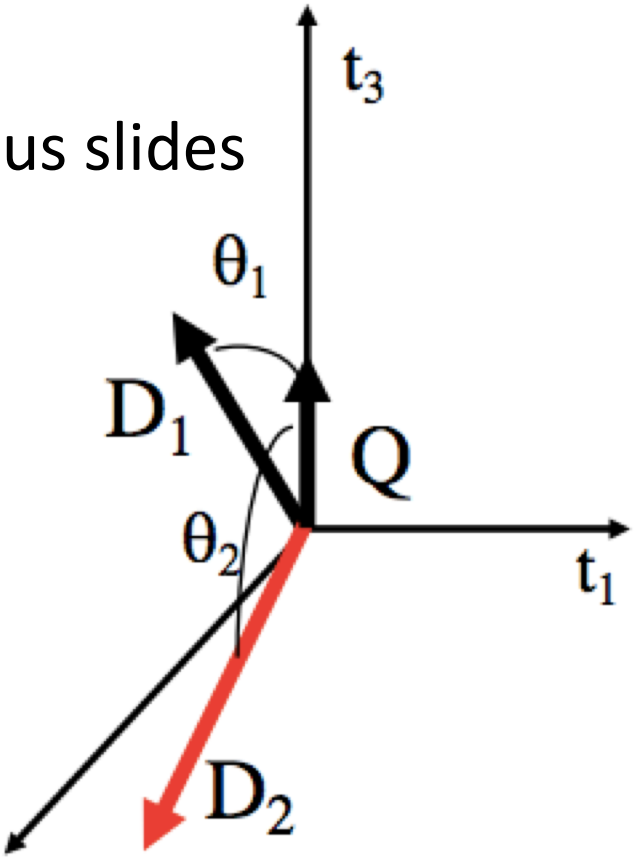
Cosine Similarity

- Normalize inner product by vector lengths
- Given two vectors, d_1 and d_2, then cosine angle θ is the angle between the two vectors
- Cosine similarity is defined as: $\cos(\theta) = \frac{d_1 \bullet d_2}{||d_1||\,||d_2||}$
  - where $||d_j|| = \sqrt{\sum_{i=1}^{t} d_{i,j}^2}$ is the vector magnitude

- Measures the cosine of the angle between two vectors
  - Two identical vectors: cosine sim = 1
  - Two orthogonal vectors: cosine sim = 0

Document Comparison Task

- Task: will a user like a new document Q, given he liked D1 but does not like D2?
- Follow calculation steps from previous slides
- Suppose we get:
  - cos(D1, Q) = 0.80
  - cos(D2, Q) = 0.20
- Conclusion:
  Q is 4 times more like D1 than D2
  So user is likely to like Q



Variations

- Other definitions of tf and idf exist
- Document vectors are not restricted to word vectors
  - n-gram vectors are possible
- Represent document profile in a similar way
  - View a set of documents as one big long document
  - View a set of documents as averages of multiple documents
  - Gather set of documents that user likes as his personal document profile
- tfidf is used in various natural language processing tasks, including text summarization and text classification

Key Ideas

- Content based recommendations (focus on text)
- Representation:
  - Bag of words
  - Documents as (word) vectors
  - Document space based on vector space model
- Algorithm:
  - tf-idf to create document profile
  - Cosine similarity to compare two documents