15. Information retrieval

15.1 What is information retrieval?

- Information retrieval (IR) is finding material (usually documents) of an *unstructured* nature (usually text), which satisfies an information need, from within a large collection.
- Science of searching for
  - information in documents
  - documents themselves
  - metadata that describe documents
  - text, sound, images, or data
- Traditionally a task of few (e.g., librarians), nowadays an every day task of millions of people (search engines).
- Semantic and statistical.

15.2 Terminology

- Retrieval
  - The task executed by the information system in response to user's request:
    - Attempt to find relevant documents to respond to user's query.
    - Match the language of the query to the language of the documents.
- Query
  - The expression of user's information need in the input language provided by the information system.
  - Most commonly keywords and Boolean connectives.
- Unstructured data
  - The data does not have semantically explicit structure.
  - Not in (easily) computer accessible format.

- Keyword
  - Pre-selected index-term that can be used to refer to the contents of the document.
- Term
  - Semantic unit: either word, sentence or a root of a word.
- Document
  - Sequence of terms
  - Unit of retrieval: paragraph, section, chapter, whole book.
- Collection (aka database)
  - Organized and relatively stationary repository of documents.

15.3 Challenges

- Given a user's query, find a set of relevant documents.
- Words may have multiple meanings.
  - “Take a picture”
  - “Take a lot of time”
  - “Take money to the bank”
- Words may appear as different parts of speech:
  - “I saw her duck.”
  - “I saw her duck away from the ball falling from the sky.”
- “The man named Abraham owned a Lincoln.”

- Measuring effectiveness
  - Find all the relevant documents, and
  - as few irrelevant ones as possible.
- Relevance
  - Is highly subjective
  - Can be measured
    - Compare several retrieval systems by the goodness of their answers to the queries judged by the users.
    - Compare several users' judgments to see how often they agree on the relevance of items retrieved by a single system.

15.4 Precision and recall

- *Recall* proportion of relevant material actually *retrieved*.
- *Precision* proportion of retrieved material actually *relevant*.
- How to measure the real number of relevant documents?

Precision = x/y

Recall = x/z

15.5 Retrieval approaches

- Semantic
  - Syntactic and semantic analysis to achieve some sort of understanding of the user's input. (Classical NLP)
  - Sense disambiguation
- Statistical methods
  - Boolean, extended boolean, vector space, and probabilistic methods.
  - Break documents and queries into terms (words) that are preprocessed.
  - The documents retrieved are the ones matching most closely the user's input in terms of some statistical measure (similarity).
  - Some sophisticated systems use phrases as terms, i.e., adjacent words that frequently co-occur in the given collection.

- N-grams
  - Documents and queries are broken into arbitrary sequences of N consecutive characters
  - For instance, moving window of N characters.
  - May or may not ignore word, phrase or punctuation boundaries.
  - Language independent.
  - Insensitive to spelling errors, typos, bad print quality.
- Weights
  - Assigned to terms in documents and the queries.
  - A given term is assigned a weight within a given document $\rightarrow$ How effective the term is in distinguishing the given document from other documents in the collection.

15.6 Preprocessing

- Removal of stop words:
  - Extremely common words, such as *the*, *a*, *and*, *but*, *it*, *you*, *in*, *at*, and forms of *be*.
  - They do not help in discrimination of relevant documents from irrelevant ones.
  - To save space, and to speed up search.
  - Stop list is language-dependent.
  - Stop word list also depends on the collection.
  - Google gives different results for the following queries:
    - to be or not to be
    - "to be or not to be."

- Stemming
  - Removal of suffixes -ED, -ING, -ION, -IONS, -BLE, -MENT, etc.
  - In order to eliminate variation due to different grammatical forms: retrieved, retrieval, retrieves, etc. should be recognized as the same word.
  - In order to improve IR systems performance, not as a linguistic exercise:
    - May produce incorrect roots: both 'fly' and 'flies' become 'fli'.
    - It is not clear when the suffix should be removed: for instance, connecting vs. thing.
  - Porter's algorithm 1980.

15.7 Classical Boolean approach

- Query formulated as a Boolean combination of terms.
- AND, OR, NOT
- Query "t_1 AND t_2" is satisfied by the given document D_1 iff D_1 contains both terms t_1 and t_2.
- Boolean query is either true or false:
  - The document either satisfies the query (is relevant), or does not satisfy (is non-relevant)
  - No ranking, no weights.
- Several possible refinements:
  - Apply query to specified component of the document: title, abstract, words in the beginning of the title, etc.
  - Proximity operator, e.g., the terms in the query must be within a certain distance from each other, in the same sentence, etc.
  - Apply proximity operator to Boolean operators: adjacent parts of the document must satisfy different Boolean conditions.

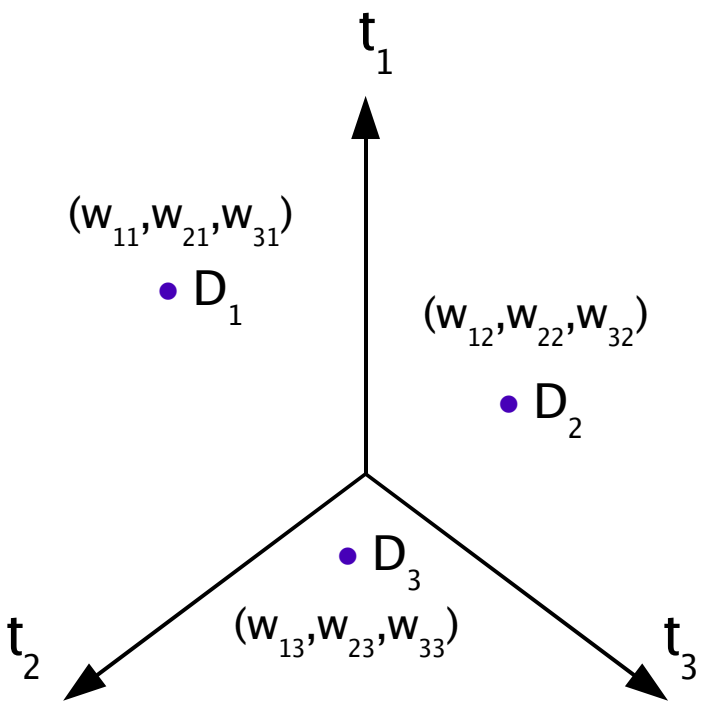
15.8 Shortcomings of Boolean approach

- Users tend to misinterpret Boolean operators AND and OR:
  - They use AND (set intersection) when OR (set union) is required.
- Boolean operators are crude, and produce counter-intuitive results:
  - OR: a document containing all of query terms is treated equally to a document that contains just one.
  - AND: a document containing all but one query term is treated as badly as a document containing none.
- *Extended boolean approaches* assign weights to terms in documents.

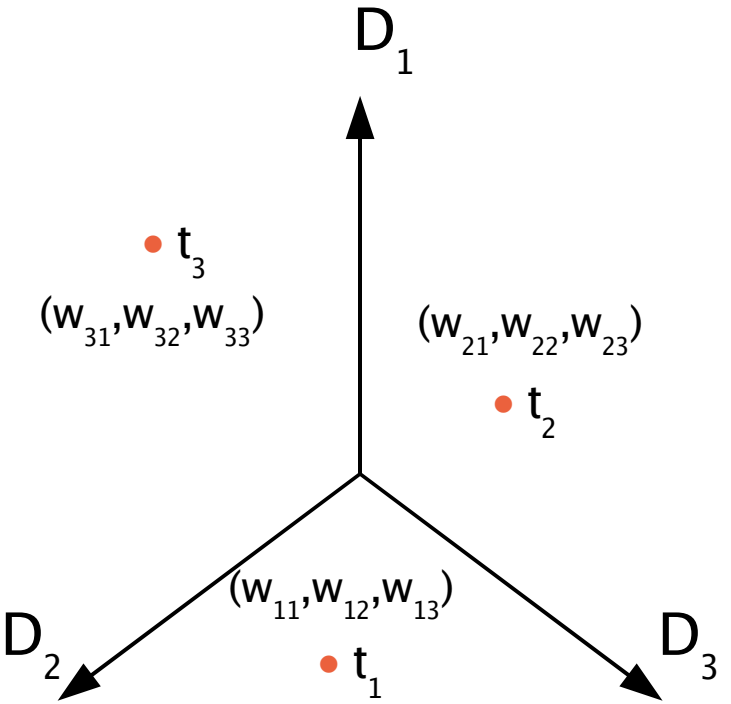
15.9 Vector space approach

- Each textual document represented as a set of terms.
- Terms are most commonly words extracted from the document $\rightarrow$ word order information is compromized.
- Union of the sets of terms represent the whole collection.
- Assign a numeric weight to each term that represents its usefulness as a descriptor of the document.
- Note, terms may have different weights in different documents; particularly, a term that does not occur in a document has a zero weight.

Document space



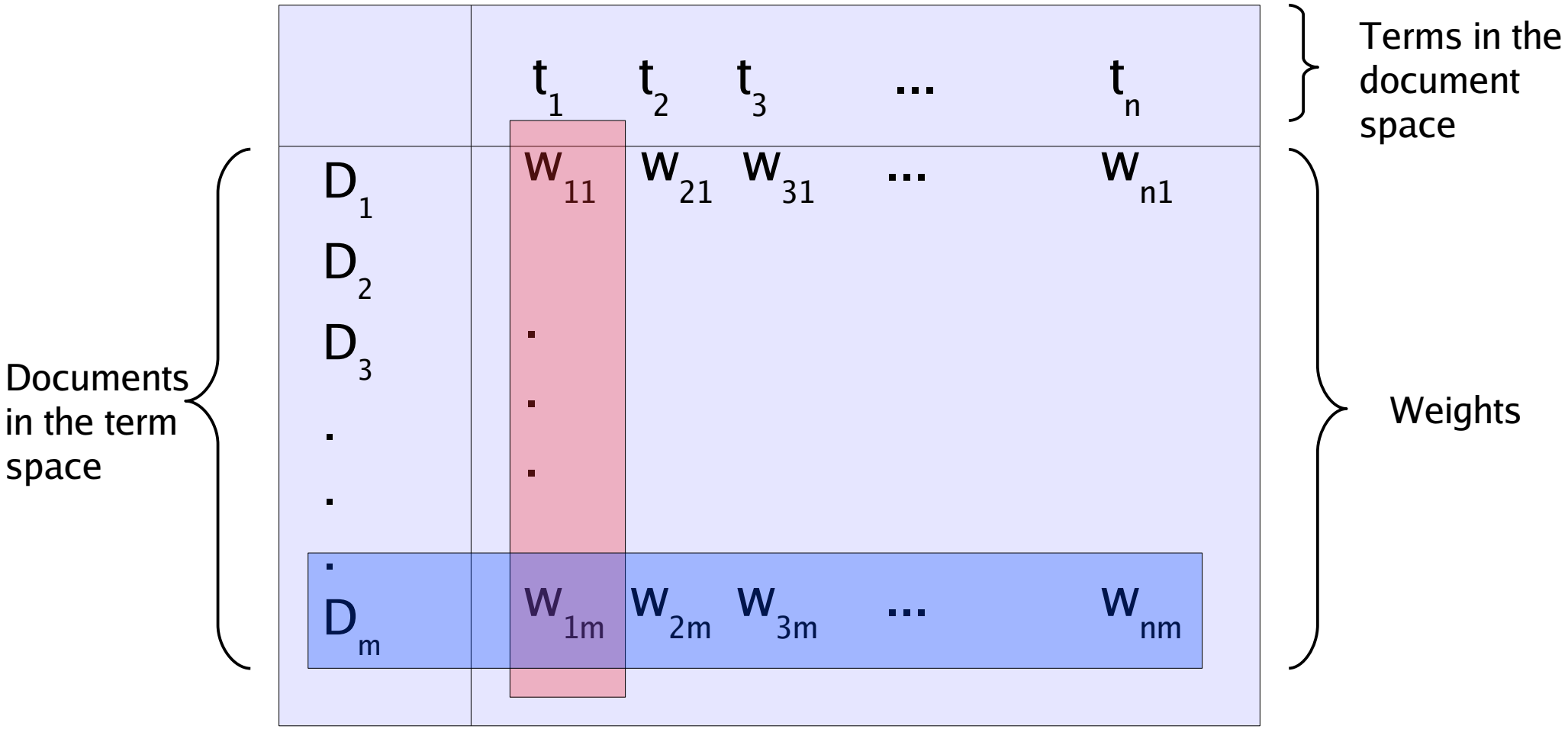
Term space for collection



w_{ij} = the weight of the term i in the document j.

15.10 Representation of collection

- Document-by-term matrix:



- How to assign weights?

15.11 Weights

- Measure how important a word is to a document in the collection:
  - Importance increases proportionally to the number of occurrences in the document.
  - Importance decreases as the number of occurrences in the whole collection increases.
- Common method: TF-IDF
- Term frequency $\mathrm{TF}(w_i, D) = n_i / \sum_k n_k$
- Inverse document frequency $\mathrm{IDF}(w_i) = \log |D| / \mathrm{DF}(w_i)$
- $\text{TF-IDF}(D, w_i) = \mathrm{TF}(w_i, D) \times \mathrm{IDF}(w_i)$
- Normalize TF to account for differences in document lengths.

# of documents in the corpus

Document frequency of w_i = # of documents in the collection containing w_i.

15.12 Example of TF-IDF

"When did an elephant walk through the streets of New York?"

- Remove stop words and suffixes:

  "Elephant walk through street New York"
- Use a corpus of documents about New York City.
- Calculate IDF values for all words in the query.
  - 'Elephant' will likely obtain a high score, and other terms ('walk', 'through', etc.) will obtain lower scores.
- Consult the index of the corpus and retrieve documents that contain all or some of the terms in the query.
  - For instance, apply Boolean query: 'Elephant' AND ('walk' OR 'through' OR 'street' OR 'New' OR 'York')
- Calculate TF to find a potentially relevant documents containing the most occurrences of 'elephant'.

15.13 Similarity

- Compute vectors for the query and the documents in the collection

  $\rightarrow$ Find numeric similarity between the query and each document.

  $\rightarrow$ Rank documents according to their similarity.
- Hopefully, the set of documents with high similarity contains many relevant documents, and the set with low similarity does not.
- Common method is the inner product: $\sum_{i=1,N} QT_i \bullet DT_i$,

  where N is the number of descriptor terms common to the query and the document, and the components of query vector QT and document vector DT are weights.

- If the vectors have been cosine normalized
  - Maximum similarity is 1; the angle between two vectors is 0.
  - Minimum similarity is 0; two vectors are orthogonal.
- Other similarity measures, i.e., distance functions, can be used:

$$L_p(D_1, D_2) = [\sum_{i=1,N} |d_{1i} - d_{2i}|^p]^{1/p}$$

  - If $p=1$, the distance is the "city block distance"
  - If $p=2$, the distance is Euclidian.
  - If $p=\infty$, the largest difference dominates.

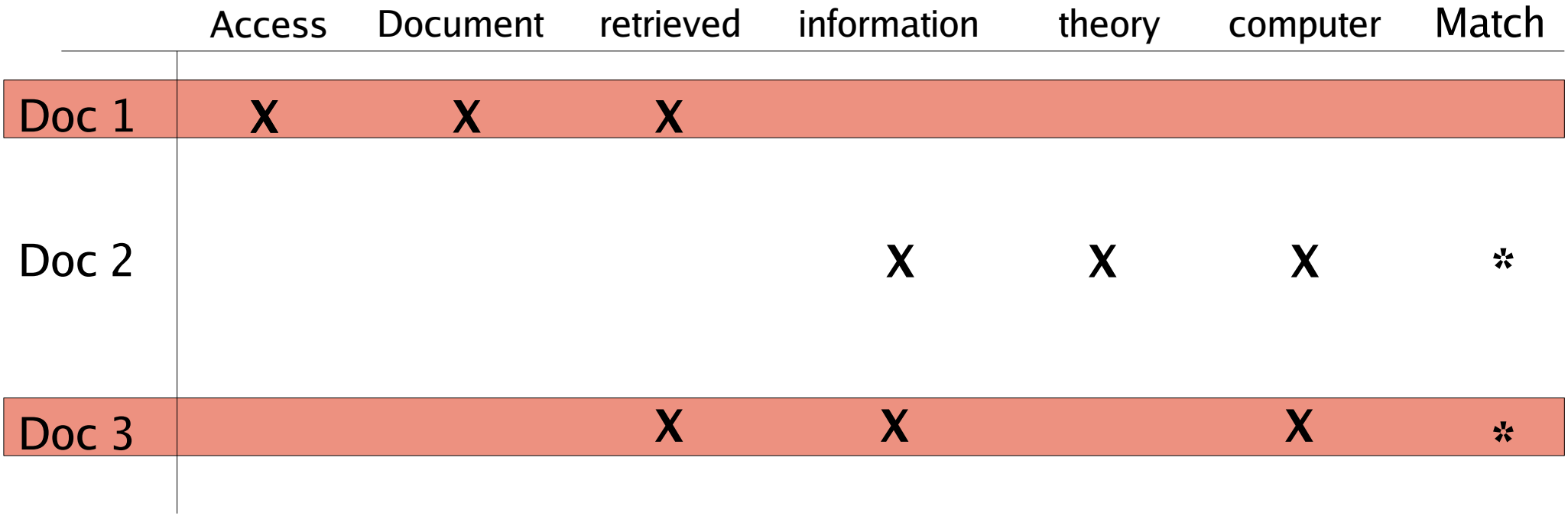
15.14 Probabilistic approach

- Serious attempt to ground the estimates of relevance in probability theory.
- Not very different from statistical methods since probabilities are based on statistical evidence.
- Conditional probability $\Pr(D|A, B, C, ...)$: probability that document D is relevant given the clues A, B, C, etc., where clues can be complex.
- Advantages and disadvantages:
  - More reliable prediction of relevance.
  - More easily understood by users (compared to, for instance, cosine similarity)
  - Better indicators of success than precision and recall.
  - Relevance probabilities based on simplifying assumptions.

15.15 Problems with current systems

- Users want to retrieve documents on the basis of conceptual content, not just based on word to document match.
- Words searchers use are different from the words used in indexing.
  - Synonymy
    - There are many ways to refer to same object.
    - User from different background, context, or with different needs describe the same information using different terms.
  - Polysemy
    - Most words have more than one distinct meaning.
    - In different context and when used by different people.
- Problem of automatic indexing

15.16 Example



Query: “IDF in computer-based information look-up”

15.17 Latent Semantic Indexing

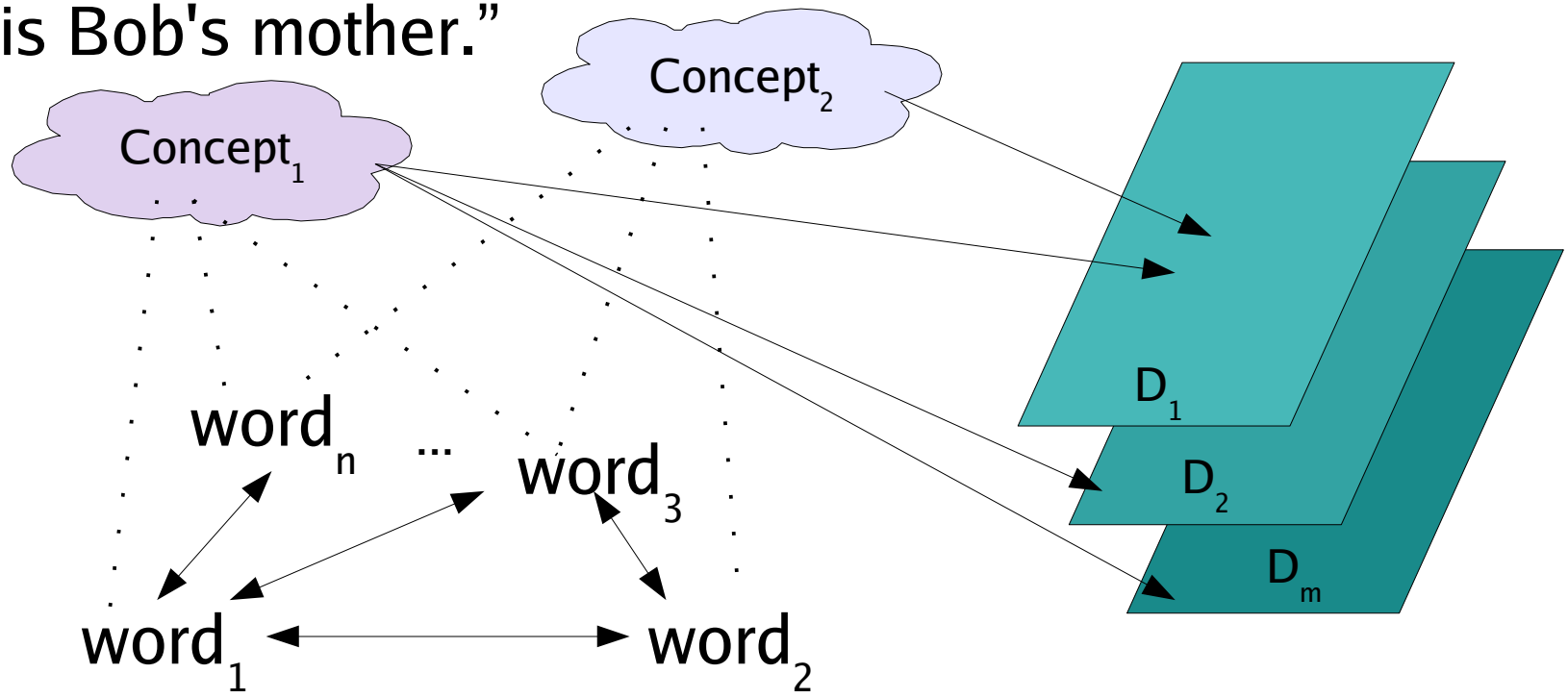
- Match queries to documents with similar topics when the query and the document use different words.
- Not based on surface level contingencies, i.e., co-occurrences of words or correlation in usage, but some “deeper” structure.
- Latent semantics
  - Occurrence of patterns of words gives a strong clue to the likely occurrence of others.
  - Based on the word use, we can predict that a given term is associated with a document even if we never observe the association.
- Does better than a system that matches terms in queries to terms in documents.

15.18 Latent Semantic Analysis

- Landauer & Dumais, 1997
- LSA is an automatic technique for extracting and inferring relations of expected contextual usage of words in passages. (Landauer *et al*., 1998)
- Twofold goal:
  - Practical tool to approximate word meaning in information retrieval.
    - Meaning of a word = an average of the meanings of all the passages it appears in.
    - Meaning of a passage = an average of the meanings of the words it contains.
  - Fundamental computational theory of the acquisition and representation of knowledge.
- Based on r*eduction of dimensionality*
  - The number of parameters in which words and passages are described.
  - Produces a better approximation of human cognitive relations.

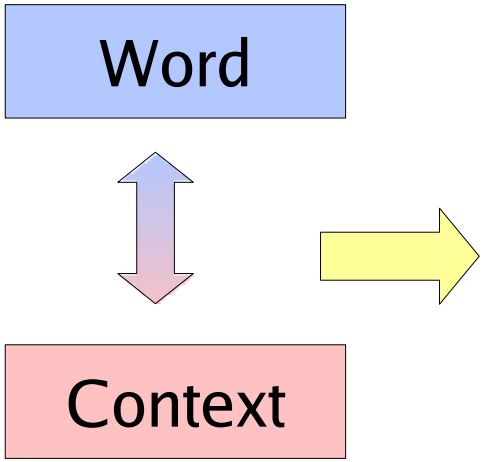
15.19 Latent semantics

- Input is a set of word-to-document relations; LSA turns them into word-to-concept and concept-to-document relations.
- LSA applied both to query and the documents.
- Example: "John is Bob's father and Mary is Ann's mother."
  - "Mary is Bob's mother."



15.20 Dimensionality reduction I

First, represent all word-to-context relations.

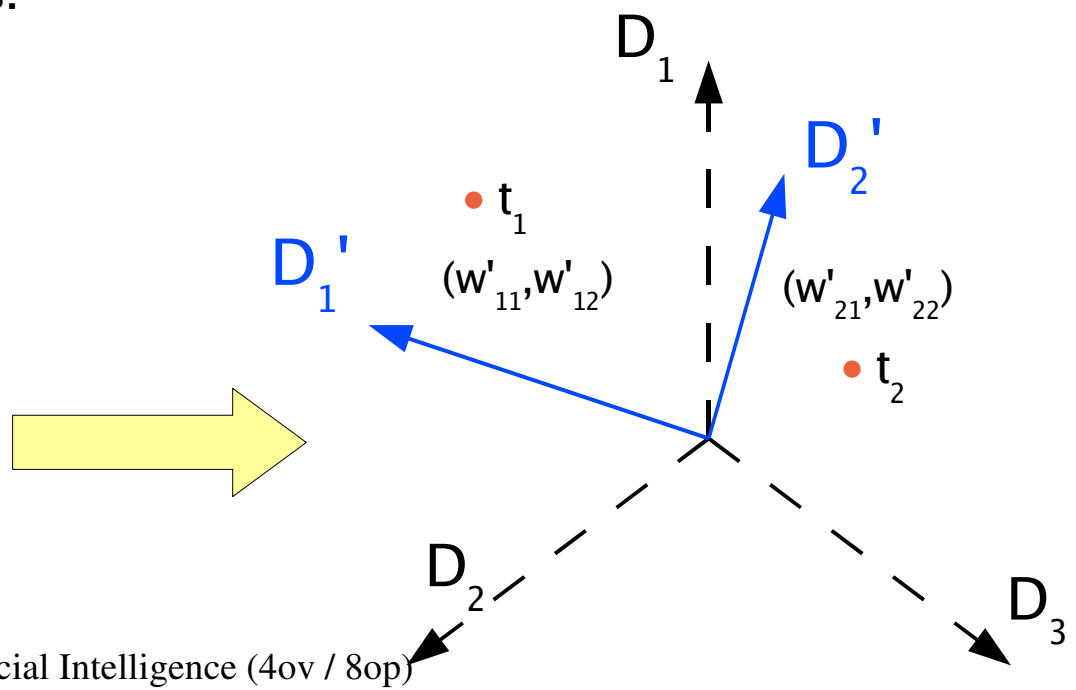


Find a set of independent components that describe the relations.

Dimensionality reduction = Abstraction that captures the mutual implications of the words and the passages.

The problem: How to find the optimal dimensions?

	t_1	t_2	t_3	...	t_n
D_1	w_{11}	w_{21}	w_{31}	...	w_{n1}
.	.				
.	.				
.	.				
D_m	w_{1m}	w_{2m}	w_{3m}	...	w_{nm}



- Reasons for dimension reduction: the original term-document matrix is
  - Often too large for computational purposes.
  - Presumed noisy; anecdotal instances of terms need to be eliminated.
  - Overly sparse; it lists only the words actually in the documents.
- Potentially takes care of
  - Synonymy — merge dimensions associated with terms with similar meanings.
  - Polysemy — components of polysemous words pointing to a right direction are added to the components of words that share the sense.

15.21 Dimensionality reduction II

- First phase:
  - Represent document as a matrix containing word counts in different contexts.
  - Each entry is subject to a transformation that weighs it
    - By the estimate of the word's importance in the context.
    - Inversely by the degree to which knowing the word's occurrence provides information about which context it is in.

TF-IDF

$\{X\} =$

	C_1	C_2	C_3	... C_n
w_1	n_{11}	n_{21}	n_{31}	... n_{n1}
.	.			
.	.			
.	.			
w_m	n_{1m}	n_{2m}	n_{3m}	... n_{nm}

Words

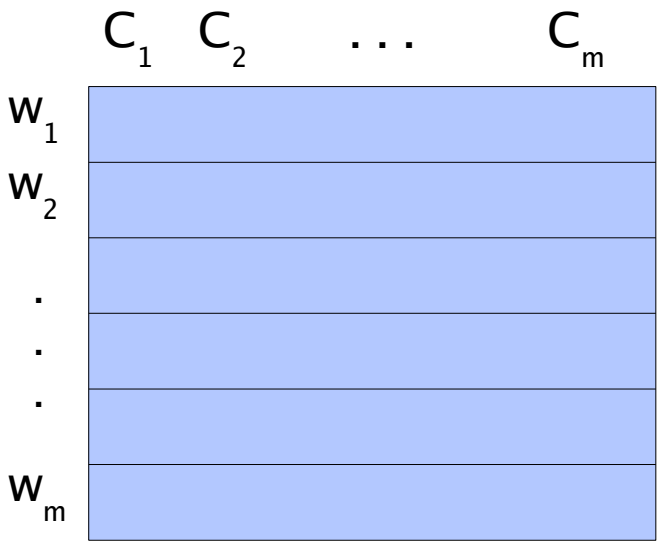
Contexts, e.g., passages

Frequency at which word in the row appears in the context of the column.

15.22 Singular value decomposition (SVD)

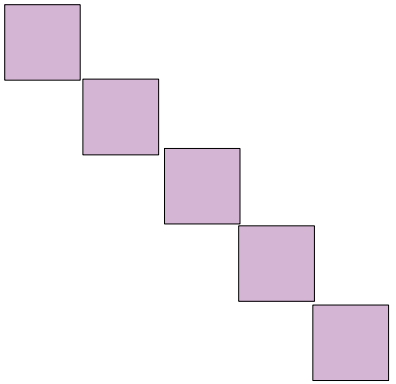
- A form of factor analysis
- Second phase: the original matrix X is decomposed into three matrices so that: {X} = {W}{S}{P}'

{W} =

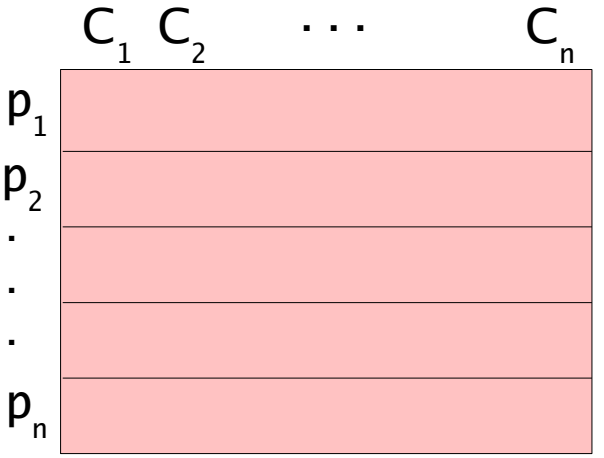


$C_i \perp C_{i+1}$ and $|C_i| = 1$, for all i.

{S} =



{P} =

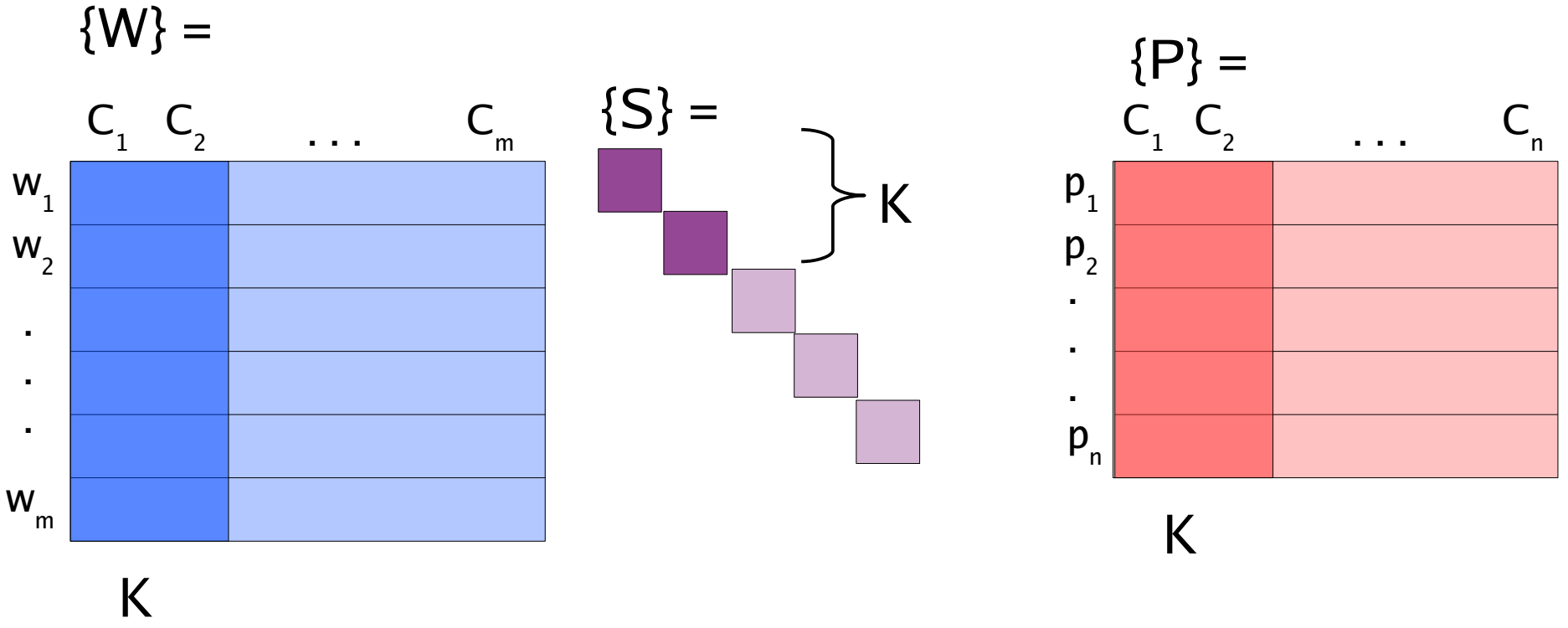


$C_j \perp C_{j+1}$ and $|C_j| = 1$, for all j.

15.23 Dimensionality reduction III

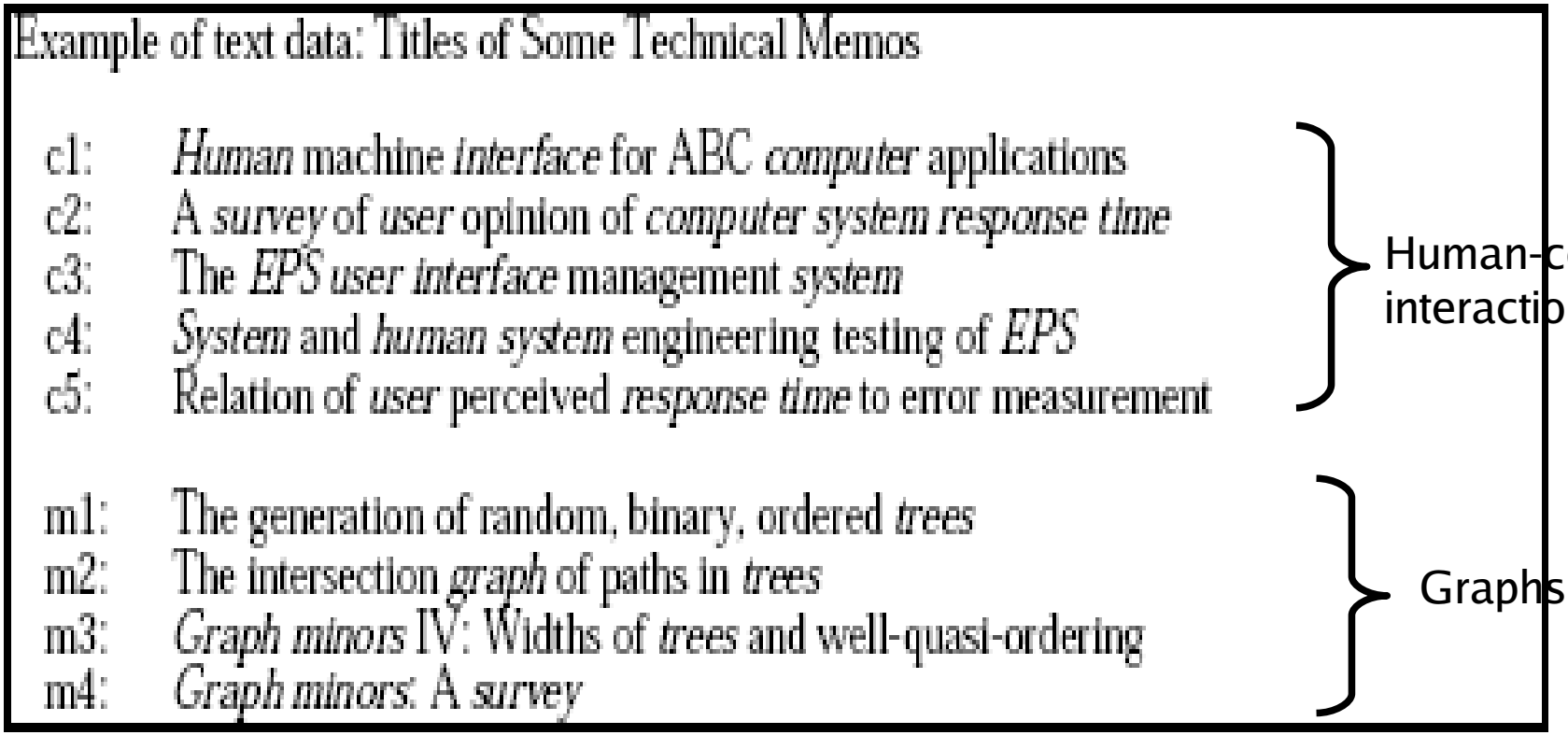
- In reconstruction phase use only the K largest singular values.

$$\{\hat{X}\} = \{W\}\{S\}\{P\}'$$



15.24 Example

- Example from Landauer *et al.*, 1998



$\{X\} =$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

A word-by-context matrix

$\{X\} = \{W\}\{S\}\{P\}'$

$\{W\} =$

0.22	-0.11	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	-0.07	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	0.04	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	0.06	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	-0.17	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	0.11	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	-0.14	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	0.27	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	0.49	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	0.62	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	0.45	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

$\{S\} =$

3.34								
	2.54							
		2.35						
			1.64					
				1.50				
					1.31			
						0.85		
							0.56	
								0.36

$\{P\} =$

0.20	0.61	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	0.17	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	-0.50	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	-0.03	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	-0.21	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	-0.26	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	-0.43	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	0.05	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	0.24	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

Singular value decomposition of X

$$\left\{\hat{X}\right\} =$$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	0.16	0.40	0.38	0.47	0.18	-0.05	-0.12	-0.16	-0.09
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	0.26	0.84	0.61	0.70	0.39	0.03	0.08	0.12	0.19
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	0.42
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	0.66
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	-0.04	0.25	-0.10	-0.21	0.15	0.22	0.50	0.71	0.62

Two-dimensional reconstruction of X; only two independent components were used. Intuitive description:

This text segment is best described as having so much of abstract concept one and so much of abstract concept two, and this word has so much of concept one and so much of concept two, and combining those two pieces of information (by vector arithmetic), my best guess is that word X actually appeared 0.6 times in context Y.

15.25 Limitations of LSA

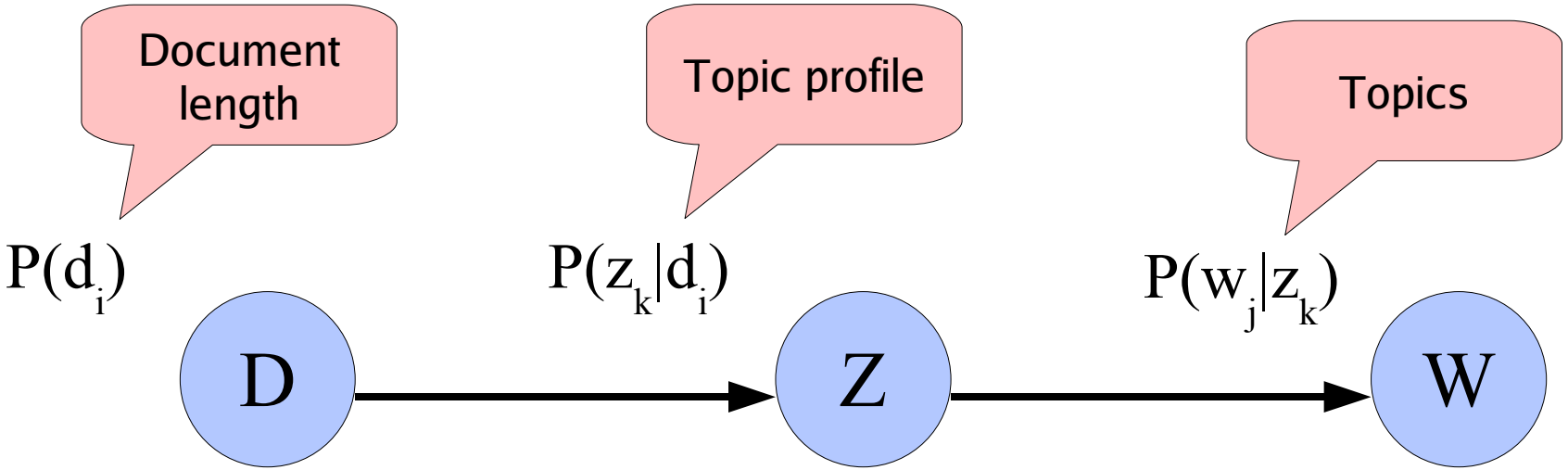
- No perception of the physical world: knowledge derived from texts only, not from the interaction with the world.
- No linguistic knowledge; does not use word order or other syntactic structure, or morphology.
- Captures the statistical regularities in processes involved in knowledge acquisition and representation, not the processes themselves.
- The performance has been demonstrated with small corpora not representative of what humans experience.
- Computational limitations to run SVD on larger corpora.

15.26 Probabilistic LSA (PLSA)

- Inspired by LSA
- Statistical model with probabilistic interpretation
  - Benefits from all the work done in statistics
    - Inference
    - Prediction
    - Model Selection
  - Computationally lighter – maybe
  - Easier to develop further

15.27 Model for word occurrences

- N documents and M different words
- K semantic topics: word distributions $P(w_j|z_k)$.
- Each document d_i has a topic profile $P(z_k|d_i)$



15.28 Learning task

- So the task is to learn the topics and the topic profiles for each document.
  - For example, in topic "spo rts", the probabilities of the words "football", and "batter" (the one hitting with baseball bat) might be significantly higher than zero, while in the topic "cooking" the probability of "foo tball" wou ld be practically zero, but the probability for "batter" (for making pancakes) would be higher.
  - A document about sports management might have the topic profile in which the topics for "sports" an d "business" are probable, but the topic for cooking is low.

15.29 Fitting the PLSA

- Calculate frequencies $n(d_i, w_j)$.
  - Stemming done and stop words removed
  - Leave "typical" words only
- Use Expectation Maximization (EM) algorithm to find the probabilities (parameters) of the model.
- Number of latent topics, K, usually picked by hand, even if something more principled could be used.

15.30 EM algorithm

- Initialize $P(w_j|z_k)$ and $P(z_k|d_i)$ – randomly or so
- While necessary:

This means "pro portional." Numbers have to be divided by their sum so that they add up to 1.

$$P(z_k|d_i, w_j) \propto P(w_j|z_k)\,P(z_k|d_i)$$

$$P(w_j|z_k) \propto \sum_{i=1}^{N} n(d_i, w_j)\,P(z_k|d_i, w_j)$$

Latent topic z_k as a word distribution

$$P(z_k|d_i) \propto \sum_{j=1}^{M} n(d_i, w_j)\,P(z_k|d_i, w_j)$$

topic profile of a document d_i

$$P(d_i) \propto N(d_i) = \sum_{j=1}^{m} n(d_i, w_j)$$

Hey, loop optimize this.

15.31 PLSI: retrieval with PLSA

- Handle query q as a document
- Calculate the topic profile of q by

$$P(z_k|q) \propto \prod_{j=1}^{M} P(w_j|z_k)^{N(q,w_j)}$$

Π denotes product

- For retrieval, compare similarities of q's topic profile and d_i's topic profiles.
  - with cosine similarity or Hellinger distance or ...
- Hoffman, T. (2001), Unsupervised Learning by Probabilistic latent Semantic Analysis. *Machine Learning, 42, 177-196*.