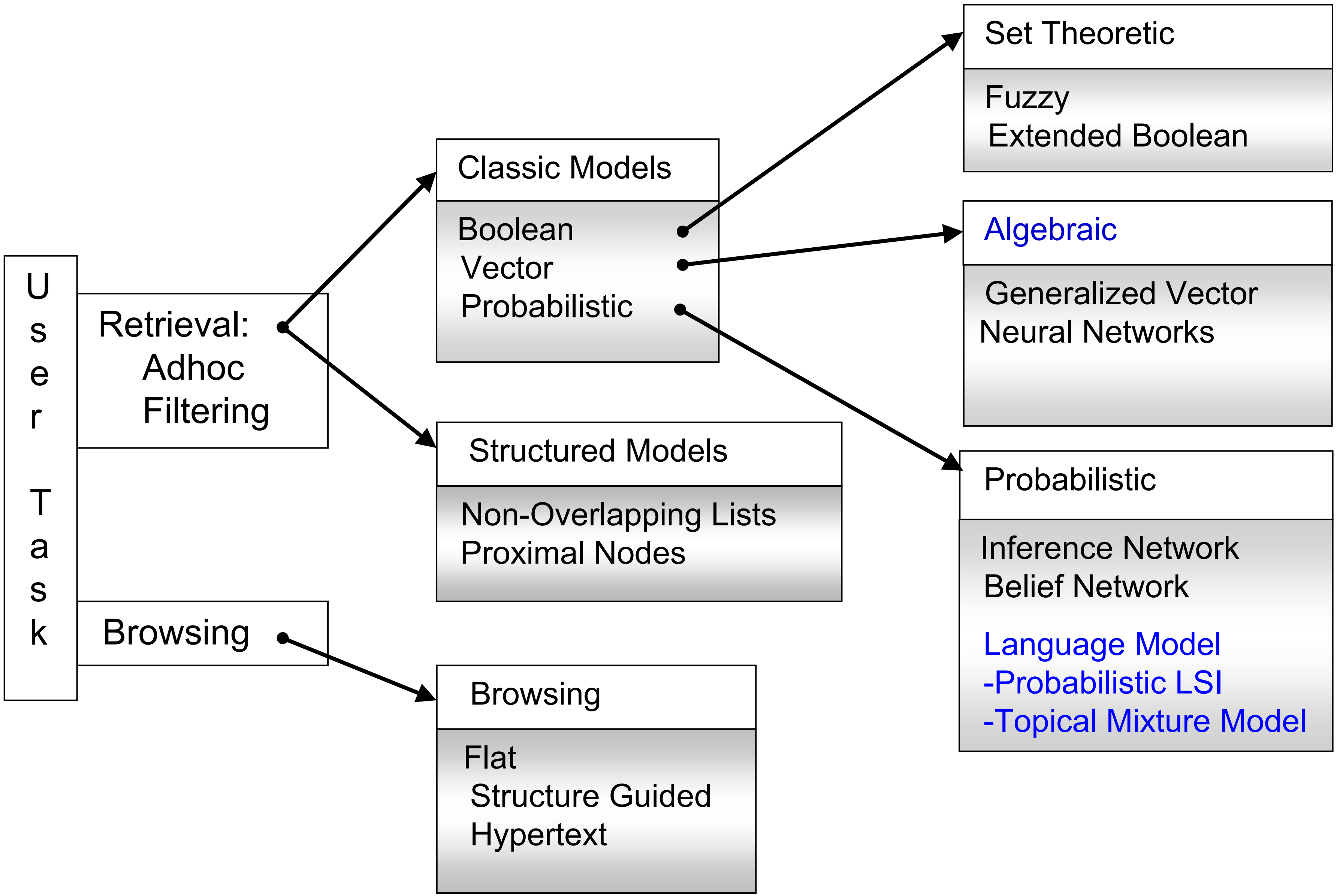
Language Models for Information Retrieval

Berlin Chen
Department of Computer Science & Information Engineering
National Taiwan Normal University

References:

1. W. B. Croft and J. Lafferty (Editors). Language Modeling for Information Retrieval. July 2003
2. T. Hofmann. Unsupervised Learning by Probabilistic Latent Semantic Analysis. Machine Learning, January-February 2001
3. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008. (Chapter 12)
4. D. A. Grossman, O. Frieder, Information Retrieval: Algorithms and Heuristics, Springer, 2004 (Chapter 2)

Taxonomy of Classic IR Models



Statistical Language Models (1/2)

- A probabilistic mechanism for "generating" a piece of text
  - Defines a distribution over all possible word sequences

$$W = w_1 w_2 \ldots w_N$$

$$P(W) = ?$$

- What is LM Used for ?
  - Speech recognition
  - Spelling correction
  - Handwriting recognition
  - Optical character recognition
  - Machine translation
  - Document classification and routing
  - Information retrieval …

Statistical Language Models (2/2)

- (Statistical) language models (LM) have been widely used for speech recognition and language (machine) translation for more than twenty years

- However, their use for use for information retrieval started only in 1998 [Ponte and Croft, SIGIR 1998]

Query Likelihood Language Models

- Documents are ranked based on Bayes (decision) rule

$$P(D|Q) = \frac{P(Q|D)P(D)}{P(Q)}$$

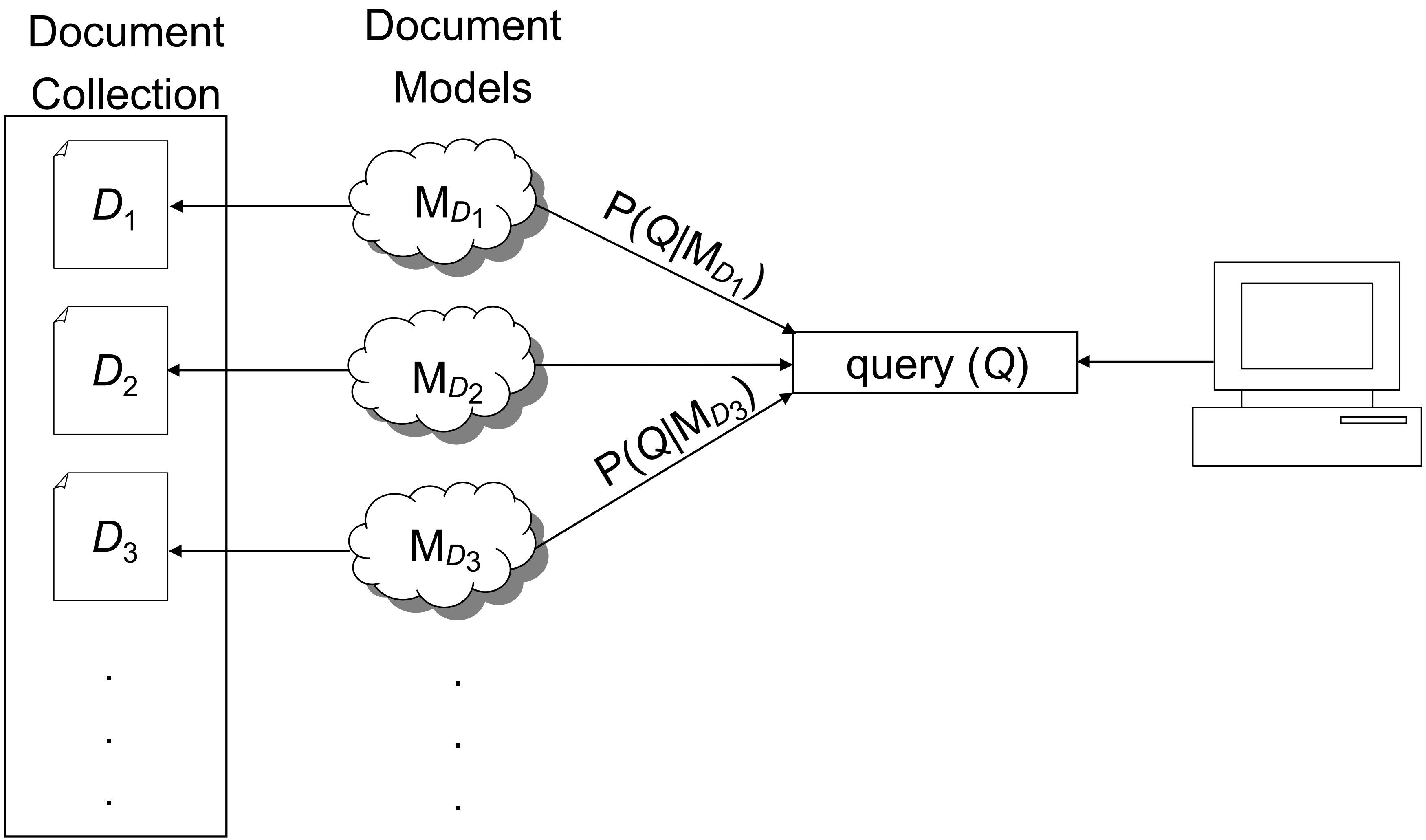
  - $P(Q)$ is the same for all documents, and can be ignored
  - $P(D)$ might have to do with authority, length, genre, etc.
    - There is no general way to estimate it
    - Can be treated as uniform across all documents

- Documents can therefore be ranked based on

$$P(Q|D) \quad \left(\text{or denoted as } P(Q|\mathrm{M}_D)\right)$$

  - The user has a prototype (ideal) document in mind, and generates a query based on words that appear in this document
  - A document D is treated as a model M_D to predict (generate) the query

Schematic Depiction



n-grams

- Multiplication (Chain) rule

$$P(w_1 w_2 \ldots. w_N) = P(w_1) P(w_2 | w_1) P(w_3 | w_1 w_2) \cdots P(w_N | w_1 w_2 \ldots w_{N-1})$$

  – Decompose the probability of a sequence of events into the probability of each successive events conditioned on earlier events

- *n*-gram assumption
  – Unigram

$$P(w_1 w_2 \ldots. w_N) = P(w_1) P(w_2) P(w_3) \cdots P(w_N)$$

    - Each word occurs independently of the other words
    - The so-called “bag-of-words” model

  – Bigram

$$P(w_1 w_2 \ldots. w_N) = P(w_1) P(w_2 | w_1) P(w_3 | w_2) \cdots P(w_N | w_{N-1})$$

  – Most language-modeling work in IR has used unigram language models
    - IR does not directly depend on the structure of sentences

Unigram Model (1/4)

- The likelihood of a query $Q = w_1 w_2 .... w_N$ given a document D

$$P(Q|\mathrm{M}_D) = P(w_1|\mathrm{M}_D)P(w_2|\mathrm{M}_D)\cdots P(w_N|\mathrm{M}_D)$$
$$= \prod_{i=1}^{N} P(w_i|\mathrm{M}_D)$$

  – Words are conditionally independent of each other given the document
  – How to estimate the probability of a (query) word given the document $P(w|\mathrm{M}_D)$?

- Assume that words follow a multinomial distribution given the document

permutation is considered here

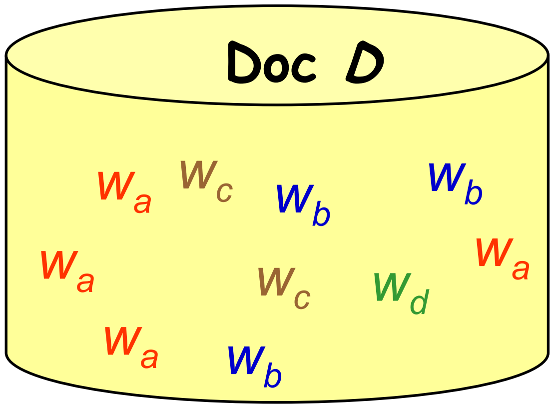
$$P(C(w_1),\ldots, C(w_V)|\mathrm{M}_D) = \frac{\left(\sum_{j=1}^{V} C(w_j)\right)!}{\prod_{i=1}^{V}(C(w_i)!)} \prod_{i=1}^{V} \lambda_{w_i}^{C(w_i)}$$

where $C(w_i)$: the number of times a word occurs

$$\lambda_{w_i} = P(w_i|\mathrm{M}_D), \quad \sum_{i=1}^{V} \lambda_{w_i} = 1$$

Unigram Model (2/4)

- Use each document itself a sample for estimating its corresponding unigram (multinomial) model
  - If Maximum Likelihood Estimation (MLE) is adopted



$$\hat{P}(w_i|\mathrm{M}_D) = \frac{C(w_i, D)}{|D|}$$

where

$C(w_i, D)$: number of times w_i occurs in D

$|D|$: length of D, $\sum_i C(w_i, D) = |D|$

$P(w_b|\mathrm{M}_D)=0.3$

$P(w_c|\mathrm{M}_D)=0.2$

$P(w_d|\mathrm{M}_D)=0.1$

$P(w_e|\mathrm{M}_D)=0.0$

$P(w_f|\mathrm{M}_D)=0.0$

The zero-probability problem

If w_e and w_f do not occur in D

then $P(w_e|\mathrm{M}_D)= P(w_f|\mathrm{M}_D)=0$

This will cause a problem in predicting the query likelihood (See the equation for the query likelihood in the preceding slide)

Unigram Model (3/4)

- Smooth the document-specific unigram model with a collection model (a mixture of two multinomials)

$$P(Q|\mathrm{M}_D) = \prod_{i=1}^{N} \left[\lambda \cdot P(w_i|\mathrm{M}_D) + (1-\lambda) \cdot P(w_i|\mathrm{M}_C) \right]$$

- The role of the collection unigram model $P(w_i|\mathrm{M}_C)$
  - Help to solve zero-probability problem
  - Help to differentiate the contributions of different missing terms in a document (global information like IDF ?)

- The collection unigram model can be estimated in a similar way as what we do for the document-specific unigram model

Unigram Model (4/4)

- An evaluation on the Topic Detection and Tracking (TDT) corpra
  - Language Model

mAP		Unigram	Unigram+Bigram
TDT2	TQ/TD	**0.6327**	0.5427
	TQ/SD	0.5658	0.4803
TDT3	TQ/TD	**0.6569**	0.6141
	TQ/SD	0.6308	0.5808

$$P_{Unigram}(Q|M_D) = \prod_{i=1}^{N}\left[\lambda \cdot P(w_i|M_D) + (1-\lambda)\cdot P(w_i|M_C)\right]$$

$$P_{Unigram+Bigram}(Q|M_D) = \prod_{i=1}^{N}\left[\lambda_1 \cdot P(w_i|M_D) + \lambda_2 \cdot P(w_i|M_C) + \lambda_3 \cdot P(w_i|w_{i-1}, M_D) + (1-\lambda_1-\lambda_2-\lambda_3)\cdot P(w_i|w_{i-1}, M_C)\right]$$

  - Vector Space Model

*m*AP		Unigram	Unigram+Bigram
TDT2	TQ/TD	0.5548	**0.5623**
	TQ/SD	0.5122	**0.5225**
TDT3	TQ/TD	0.6505	**0.6531**
	TQ/SD	0.6216	0.6233

Maximum Mutual Information

- Documents can be ranked based their mutual information with the query

$$MI(Q,D) = \log \frac{P(Q,D)}{P(Q)P(D)}$$
$$= \log P(Q|D) - \underbrace{\log P(Q)}$$

being the same for all documents, and hence can be ignored

- Document ranking by mutual information (MI) is equivalent that by likelihood

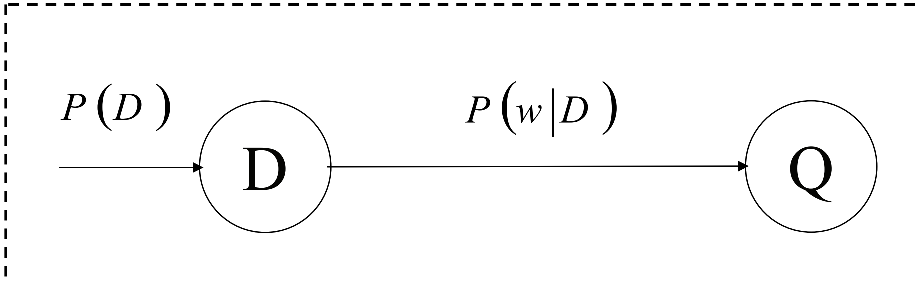
$$\arg\max_{D} MI(Q,D) = \arg\max_{D} P(Q|D)$$

Probabilistic Latent Semantic Analysis (PLSA)

Thomas Hofmann 1999

- Also called The Aspect Model, Probabilistic Latent Semantic Indexing (PLSI)
  - Graphical Model Representation (a kind of Bayesian Networks)

Language model



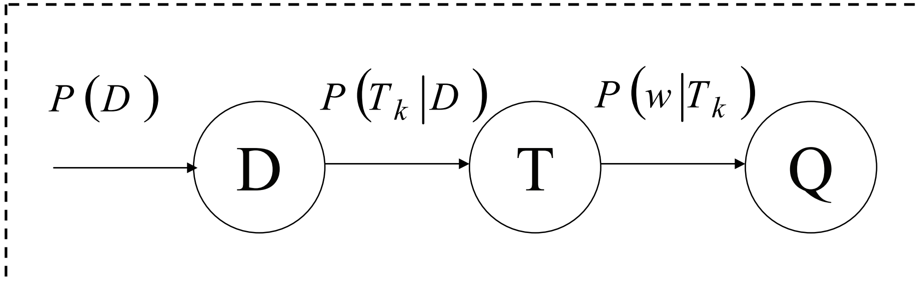
$$sim(Q,D) = P(D|Q) = \frac{P(Q|D)P(D)}{P(Q)}$$

$$\propto P(Q|D)P(D)$$

$$\approx P(Q|D)$$

$$= \prod_{w \in Q} \left[\lambda \cdot P(w|M_D) + (1-\lambda) \cdot P(w|M_C)\right]^{C(w,Q)}$$

PLSA



$$sim(Q,D) = P(Q|D) = \prod_{w \in Q} P(w|D)^{C(w,Q)}$$

$$= \prod_{w \in Q} \left[\sum_{k=1}^{K} P(w,T_k|D)\right]^{C(w,Q)}$$

$$= \prod_{w \in Q} \left[\sum_{k=1}^{K} P(w|T_k)P(T_k|D)\right]^{C(w,Q)}$$

The latent variables
=>The unobservable class variables T_k
(topics or domains)

Reference:

1. T. Hofmann. *Unsupervised Learning by Probabilistic Latent Semantic Analysis*. Machine Learning, January-February 2001

PLSA: Formulation

- Definition
  - $P(D)$: the prob. when selecting a doc D
  - $P(T_k|D)$: the prob. when pick a latent class T_k for the doc D
  - $P(w|T_k)$: the prob. when generating a word w from the class T_k

PLSA: Assumptions

- **Bag-of-words**: treat docs as *memoryless* source, words are generated independently

$$sim(Q,D) = P(Q|D) = \prod_w P(w|D)^{C(w,Q)}$$

- **Conditional independent**: the doc D and word w are independent conditioned on the state of the associated latent variable T_k

$$P(w,D|T_k) \approx P(w|T_k)P(D|T_k)$$

$$\begin{aligned} P(w|D) &= \sum_{k=1}^{K} P(w,T_k|D) = \sum_{k=1}^{K} \frac{P(w,D,T_k)}{P(D)} = \sum_{k=1}^{K} \frac{P(w,D|T_k)P(T_k)}{P(D)} \\ &= \sum_{k=1}^{K} \frac{P(w|T_k)P(D|T_k)P(T_k)}{P(D)} = \sum_{k=1}^{K} \frac{P(w|T_k)P(T_k,D)}{P(D)} \\ &= \sum_{k=1}^{K} P(w|T_k)P(T_k|D) \end{aligned}$$

PLSA: Training (1/2)

- Probabilities are estimated by maximizing the collection likelihood using the Expectation-Maximization (EM) algorithm

$$L_C = \sum_D \sum_w C(w, D) \log P(w|D)$$

$$= \sum_D \sum_w C(w, D) \log \left[\sum_{T_k} P(w|T_k) P(T_k|D) \right]$$

EM tutorial:

- Jeff A. Bilmes "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," U.C. Berkeley TR-97-021

PLSA: Training (2/2)

- E (expectation) step

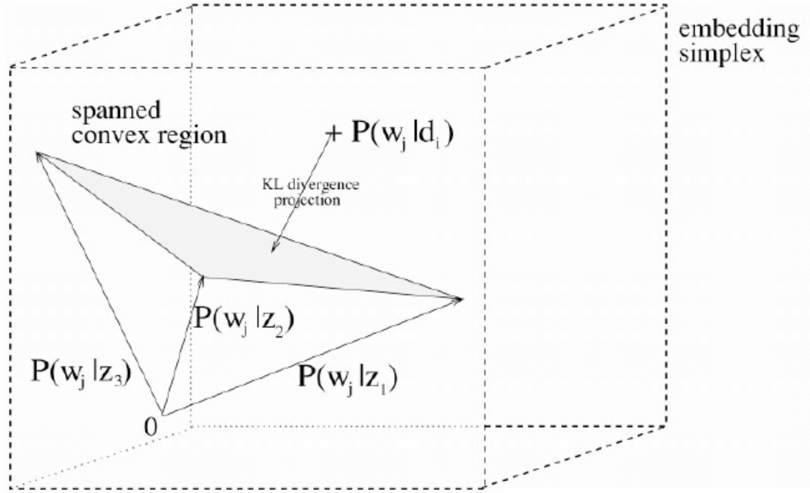
$$P(T_k|w,D) == \frac{P(w|T_k)P(T_k|D)}{\sum_{T_k} P(w|T_k)P(T_k|D)}$$

- M (Maximization) step

$$\hat{P}(w|T_k) = \frac{\sum_D C(w,D)P(T_k|w,D)}{\sum_w \sum_D C(w,D)P(T_k|w,D)}$$

$$\hat{P}(T_k|D_i) = \frac{\sum_w C(w,D)P(T_k|w,D)}{\sum_{w'} C(w',D)}$$

PLSA: Latent Probability Space (1/2)



Sketch of the probability simplex and a convex region spanned by class-conditional probabilities in the aspect model.

Dimensionality *K*=128 (latent classes)

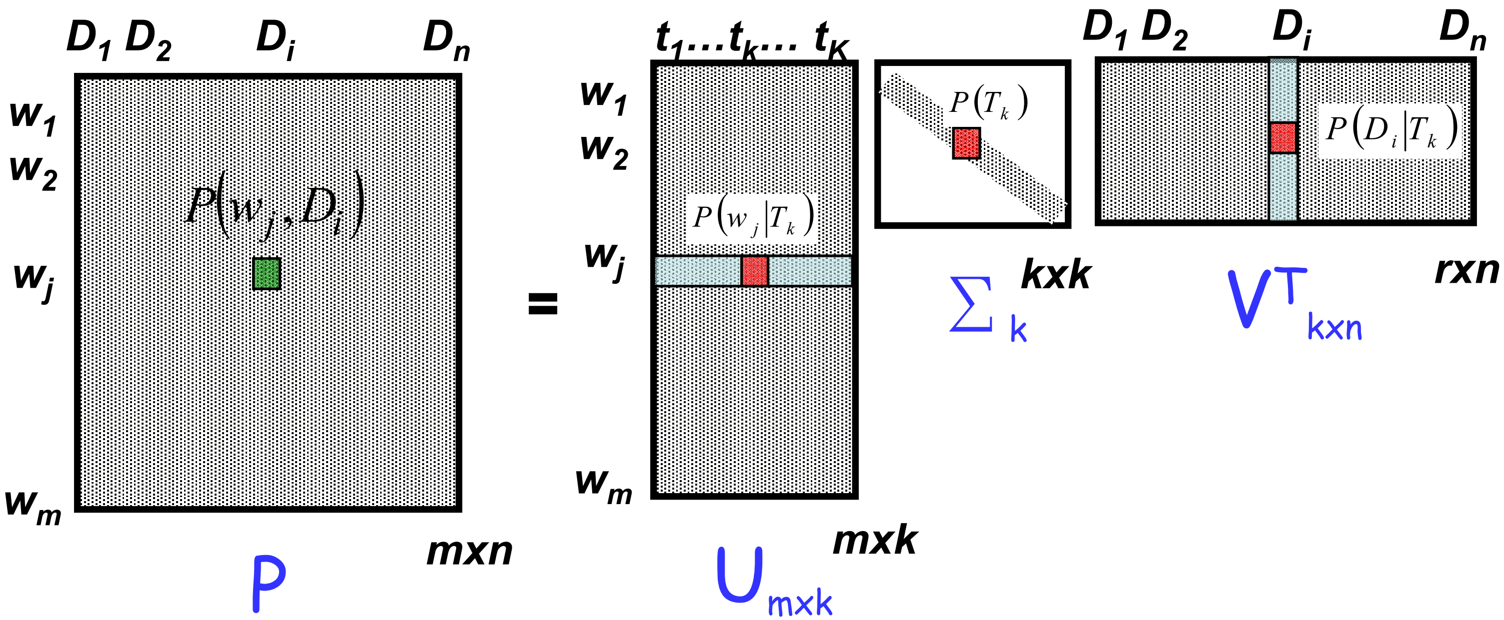
Aspect 1	Aspect 2	Aspect 3	Aspect 4
imag	video	region	speaker
SEGMENT	sequenc	contour	speech
textur	motion	boundari	recogni
color	frame	descrip	signal
tissu	scene	imag	train
brain	SEGMENT	SEGMENT	hmm
slice	shot	precis	sourc
cluster	imag	estim	speakerindepend
mri	cluster	pixel	SEGMENT
algorithm	visual	paramet	sound
medical imaging	image sequence analysis	context of contour boundary detection	phonetic segmentation

$$P(w_j, D_i) = \sum_{T_k} P(w_j, T_k, D_i) = \sum_{T_k} P(w_j | T_k, D_i) P(T_k, D_i)$$

$$= \sum_{T_k} P(w_j | T_k) P(T_k) P(D_i | T_k)$$

$$\boldsymbol{P}(\boldsymbol{W}, \boldsymbol{D}) = \hat{U} : \left(P(w_j | T_k)\right)_{j,k} \cdot \hat{\Sigma} : \mathrm{diag}\left(P(T_k)\right)_k \cdot \hat{V} : \left(P(D_i | T_k)\right)_{i,k}$$

PLSA: Latent Probability Space (2/2)



$$P(w_j, D_i) = \sum_{T_k} P(w_j|T_k)P(T_k)P(D_i|T_k)$$

PLSA: One more example on TDT1 dataset

aviation	space missions	family love	Hollywood love
Aspect 1	Aspect 2	Aspect 3	Aspect 4
plane	space	home	film
airport	shuttle	family	movie
crash	mission	like	music
flight	astronauts	love	new
safety	launch	kids	best
aircraft	station	mother	hollywood
air	crew	life	love
passenger	nasa	happy	actor
board	satellite	friends	entertainment
airline	earth	cnn	star

The 2 aspects to most likely generate the word 'flight' (left) and 'love' (right), derived from a $K = 128$ aspect model of the TDT1 document collection. The displayed terms are the most probable words in the class-conditional distribution $P(w_j | z_k)$, from top to bottom in descending order.

PLSA: Experiment Results (1/4)

- Experimental Results
  - Two ways to smoothen empirical distribution with PLSA
    - Combine the cosine score with that of the vector space model (so does LSA)

      PLSA-U* (See next slide)
    - Combine the multinomials individually

      PLSA-Q*

$$P_{PLSA}(w|D)=\sum_{k=1}^{K}P(w|T_k)P(T_k|D)$$

$$P_{PLSA-Q*}(w|D)=\lambda\cdot P_{Empirical}(w|D)+(1-\lambda)\cdot P_{PLSA}(w|D)$$

$$P_{Empirical}(w|D)=\frac{c(w,D)}{c(D)}$$

$$P_{PLSA-Q*}(Q|D)=\prod_{w\in Q}\left(\lambda\cdot P_{Empirical}(w|D)+(1-\lambda)\cdot P_{PLSA}(w|D)\right)^{c(w,D)}$$

Both provide almost identical performance

– It's not known if PLSA was used alone

PLSA: Experiment Results (2/4)

PLSA-U*

- Use the low-dimensional representation $P(T_k \mid Q)$ and $P(T_k \mid D)$ (be viewed in a *k*-dimensional latent space) to evaluate relevance by means of cosine measure
- Combine the cosine score with that of the vector space model
- Use the ad hoc approach to re-weight the different model components (dimensions) by

$$R_{PLSA-U^*}(Q,D) = \frac{\sum_k P(T_k|Q)P(T_k|D)}{\sqrt{\sum_k P(T_k|Q)^2}\sqrt{\sum_k P(T_k|D)^2}}$$

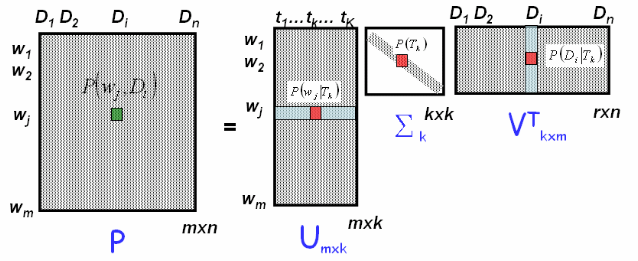
$$\text{, where } P(T_k|Q) = \frac{\sum_{w \in Q} C(w,Q)\,P(T_k|w,Q)}{\sum_{w' \in Q} C(w',Q)}$$

online folded-in

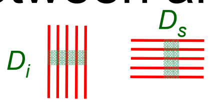
$$\tilde{R}_{PLSA-U^*}(Q,D) = \lambda \cdot R_{PLSA-U^*}(Q,D) + (1-\lambda) \cdot R_{VSM}(\vec{Q},\vec{D})$$

PLSA: Experiment Results (3/4)

- Why $R_{PLSI-Q^*}(Q,D_i)=\dfrac{\sum_k P(T_k|Q)P(T_k|D_i)}{\sqrt{\sum_k P(T_k|Q)^2}\sqrt{\sum_k P(T_k|D_i)^2}}$?



– Reminder that in LSA, the relations between any two docs can be formulated as



$A'^TA'=(U'\Sigma'V'^T)^T{}'(U'\Sigma'V'^T)=V'\Sigma'^TU^TU'\Sigma'V'^T=(V'\Sigma')(V'\Sigma')^T$

$$sim(D_i,D_s)=coine(\hat{D}_i\Sigma,\hat{D}_s\Sigma)=\frac{\hat{D}_i\Sigma^2\hat{D}_s^T}{|\hat{D}_i\Sigma||\hat{D}_s\Sigma|}$$

– PLSA mimics LSA in similarity measure

$\hat{D}_i$ and $\hat{D}_s$ are row vectors

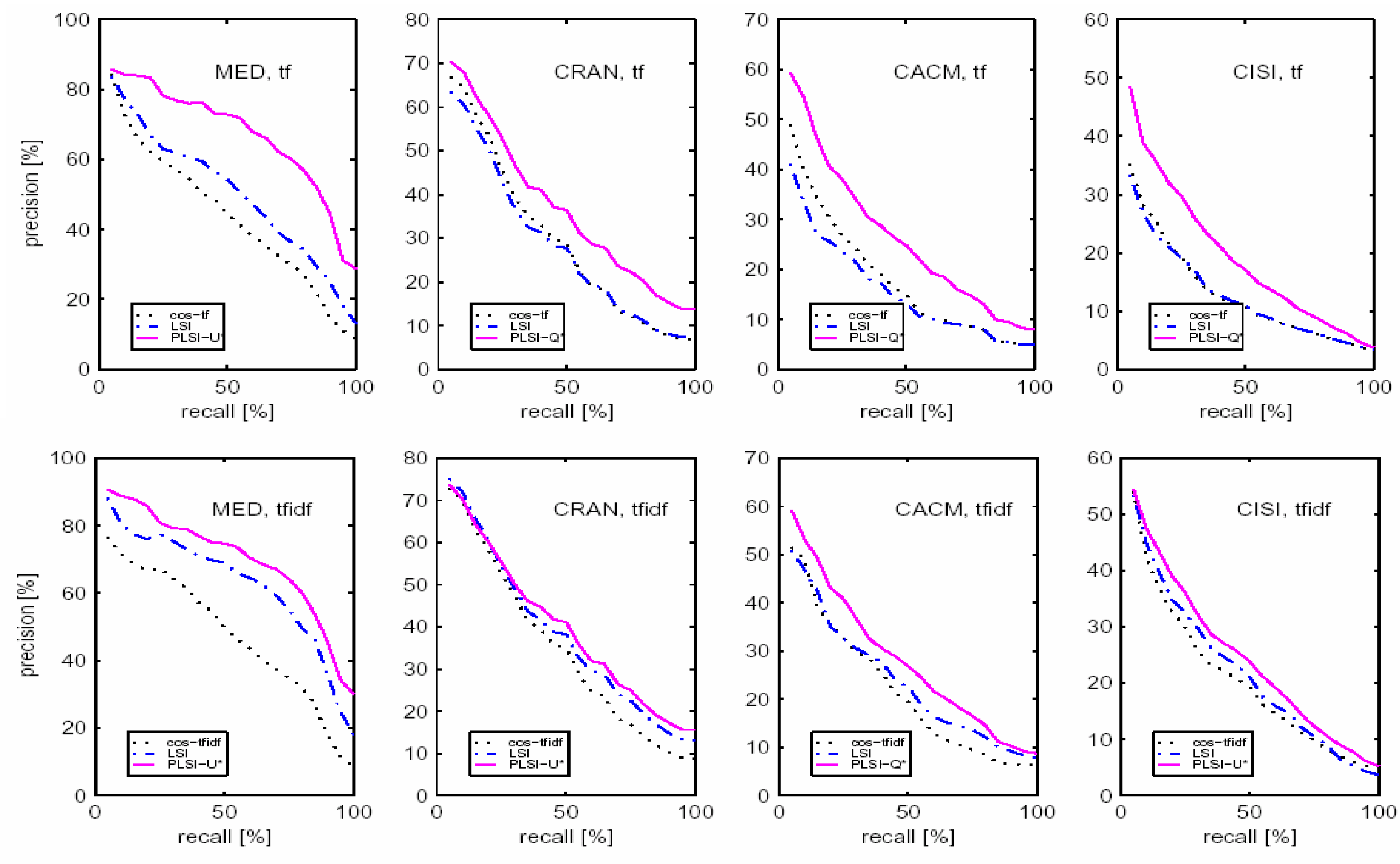
$$R_{PLSI-Q^*}(D_i,D_s)=\frac{\sum_k P(D_i|T_k)P(T_k)P(T_k)P(D_s|T_k)}{\sqrt{\sum_k[P(D_i|T_k)P(T_k)]^2}\sqrt{\sum_k[P(D_i|T_k)P(T_k)]^2}}$$

$$=\frac{\sum_k P(T_k|D_i)P(D_i)P(T_k|D_s)P(D_s)}{\sqrt{\sum_k[P(T_k|D_i)P(D_i)]^2}\sqrt{\sum_k[P(T_k|D_s)P(D_s)]^2}}$$

$$=\frac{\sum_k P(T_k|D_i)P(T_k|D_s)}{\sqrt{\sum_k P(T_k|D_i)^2}\sqrt{\sum_k P(T_k|D_s)^2}}$$

$$P(D_i|T_k)P(T_k)=P(T_k|D_i)P(D_i)$$

PLSA: Experiment Results (4/4)



PLSA vs. LSA

- Decomposition/Approximation
  - **LSA**: least-squares criterion measured on the L2- or Frobenius norms of the word-doc matrices
  - **PLSA**: maximization of the likelihoods functions based on the cross entropy or Kullback-Leibler divergence between the empirical distribution and the model

- Computational complexity
  - LSA: SVD decomposition
  - PLSA: EM training, is time-consuming for iterations ?