Chapter 4: Advanced IR Models

4.2.1 What is a Statistical Language Model?

generative model for word sequence
(generates probability distribution of word sequences,
or bag-of-words, or set-of-words, or structured doc, or ...)

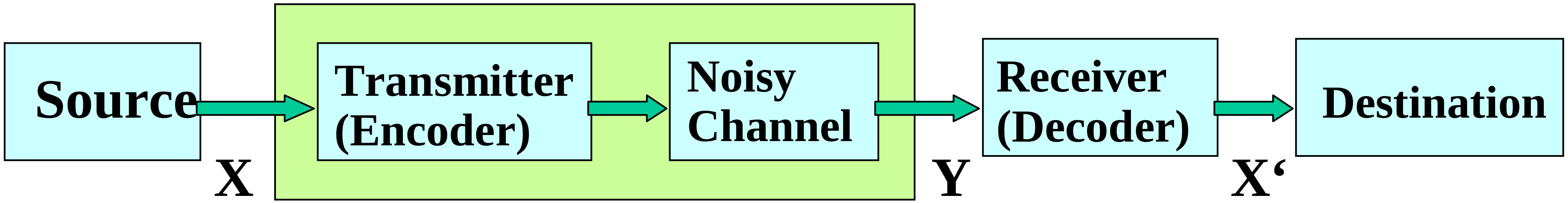
Example: P[„Today is Tuesday“] = 0.001
P[„Today Wednesday is“] = 0.000000001
P[„The Eigenvalue is positive“] = 0.000001

LM itself highly context- / application-dependent

Examples:

- **speech recognition**: given that we heard „Julia“ and „feels“, how likely will we next hear „happy“ or „habit“?
- **text classification**: given that we saw „soccer“ 3 times and „game“ 2 times, how likely is the news about sports?
- **information retrieval**: given that the user is interested in math, how likely would the user use „distribution“ in a query?

Source-Channel Framework [Shannon 1948]



P[X] **P[Y|X]** **P[X|Y]=?**

$$\hat{X} = \arg\max_X P[X \mid Y] = \arg\max_X P[Y \mid X] P[X]$$

X is text $\rightarrow$ P[X] is language model

<u>Applications:</u>

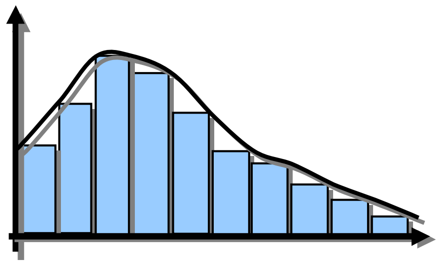
speech recognition	X: word sequence	Y: speech signal
machine translation	X: English sentence	Y: German sentence
OCR error correction	X: correct word	Y: erroneous word
summarization	X: summary	Y: document
information retrieval	X: document	Y: query

Text Generation with (Unigram) LM

LM θ: P[word | θ] —— sample ——> document d

LM for topic 1: IR&DM

...	
text	0.2
mining	0.1
n-gram	0.01
cluster	0.02
...	
food	0.000001

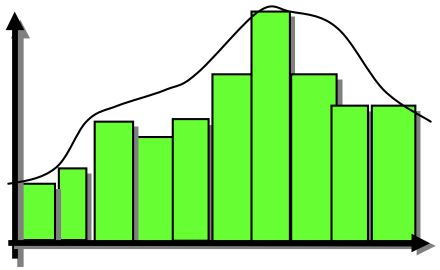


text
mining
paper

different θ_d for different d

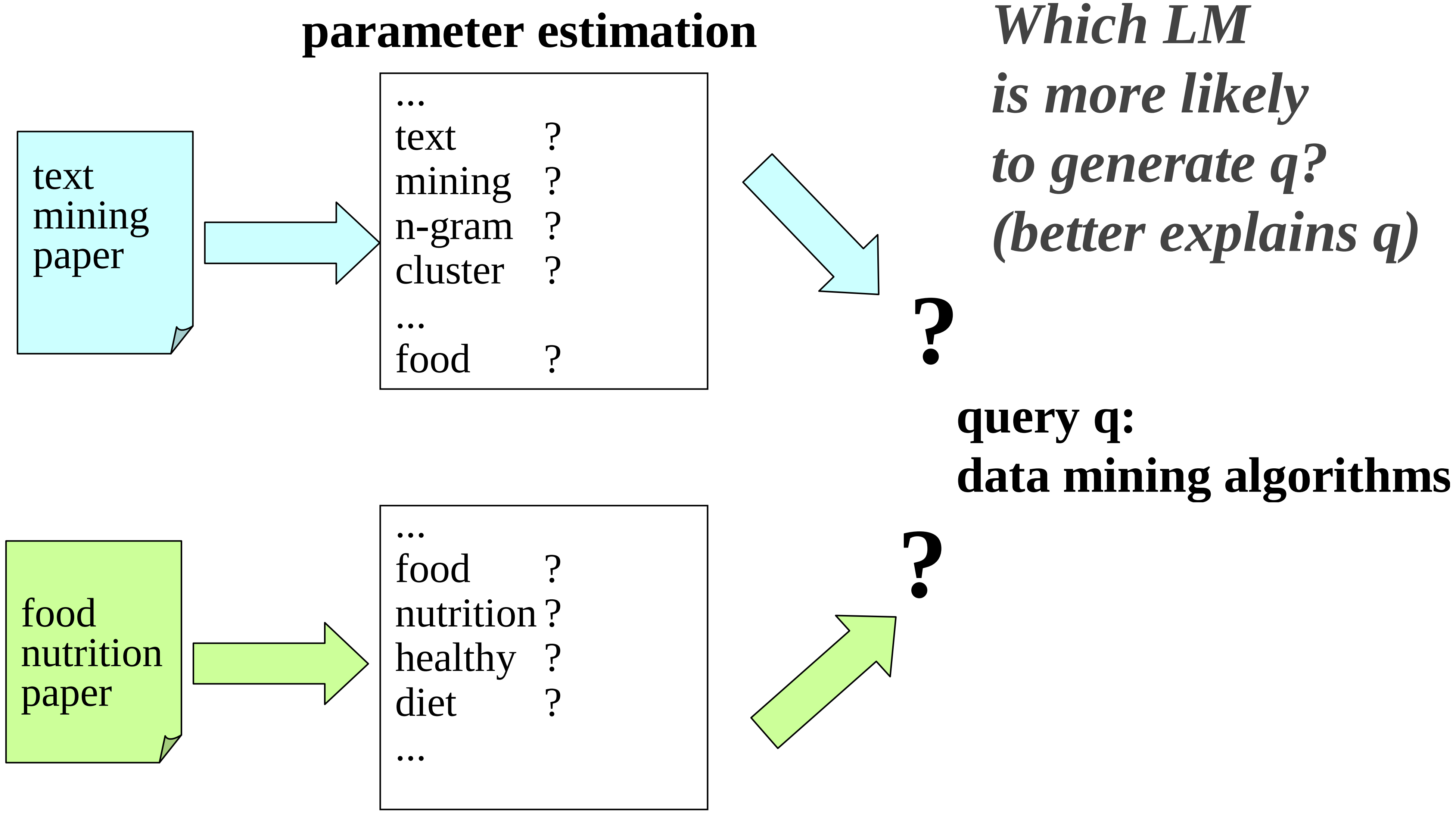
LM for topic 2: Health

...	
food	0.25
nutrition	0.1
healthy	0.05
diet	0.02
...	



food
nutrition
paper

Basic LM for IR



IR as LM Estimation

P[R|d,q]

user likes doc (R)
given that it has features d
and user poses query q

$$\sim \frac{P[d \mid R, q]}{P[d \mid \overline{R}, q]}$$

prob. IR

$$\sim P[q, d \mid R] \cdot P[R]$$

$$= P[q \mid d, R] \cdot P[d \mid R] \cdot P[R]$$

$$\sim P[q \mid d]$$

statist. LM

query likelihood:

$$s(q,d) = \log P[q \mid d] = \sum_{j \in q} P[j \mid \theta_d]$$

top-k query result:

$$k - \arg\max_d \log P[q \mid d]$$

MLE would be tf

Multi-Bernoulli vs. Multinomial LM

Multi-Bernoulli:

$$P[q \mid d] = \Pi_{j \in q}\, p_j(d)^{X_j(q)} \cdot (1 - p_j(d))^{1 - X_j(q)}$$

with Xj(q)=1 if j∈ q, 0 otherwise

Multinomial:

$$P[q \mid d] = \binom{|q|}{f(j_1)\, f(j_2) \ldots f(j_{|q|})} \Pi_{j \in q}\, p_j(d)^{f_j(q)}$$

with f_j(q) = f(j) = relative frequency of j in q

multinomial LM more expressive and usually preferred

LM Scoring by Kullback-Leibler Divergence

$$\log_2 P[q \mid d] = \log_2 \binom{|q|}{f(j_1)\, f(j_2) \ldots f(j_{|q|})} \Pi_{j \in q}\, p_j(d)^{f_j(q)}$$

$$\sim \Sigma_{j \in q}\, f_j(q) \log_2 p_j(d)$$

$$= -H(f(q), p(d))$$

neg. cross-entropy

$$\sim -H(f(q), p(d)) + H(f(q))$$

$$= -D(f(q) \parallel p(d))$$

$$= -\sum_j f_j(q) \log_2 \frac{f_j(q)}{p_j(d)}$$

neg. KL divergence of θ_q and θ_q

4.2.2 Smoothing Methods

absolutely crucial to avoid overfitting and make LMs useful (one LM per doc, one LM per query !)

possible methods:

- **Laplace smoothing**
- **Absolute Discouting**
- **Jelinek-Mercer smoothing**
- **Dirichlet-prior smoothing**
- **...**

most with their own parameters

choice and parameter setting still pretty much black art (or empirical)

Laplace Smoothing and Absolute Discounting

estimation of θ_d: $p_j(d)$ by MLE would yield $\frac{freq(j,d)}{|d|}$

where $|d| = \sum_j freq(j,d)$

Additive Laplace smoothing:

$$\hat{p}_j(d) = \frac{freq(j,d)+1}{|d|+2}$$

Absolute discounting:

$$\hat{p}_j(d) = \frac{\max(freq(j,d)-\delta,0)}{|d|} + \sigma \frac{freq(j,C)}{|C|}$$

with corpus C, $\delta \in [0,1]$

where $\sigma = \frac{\delta \cdot \#distinct\ terms\ in\ d}{|d|}$

Jelinek-Mercer Smoothing

<u>Idea:</u>
use linear combination of doc LM with background LM (corpus, common language);

$$\hat{p}_j(d) = \lambda \frac{freq(j,d)}{|d|} + (1-\lambda) \frac{freq(j,C)}{|C|}$$

could also consider query log as background LM for query

Dirichlet-Prior Smoothing

$$\hat{p}_j(d) = \left(\lambda P[j\,|\,d] + (1-\lambda) P[j\,|\,C]\right)$$

tf

$$= \frac{|d|\cdot P[j\,|\,d]}{|d|+s} + \frac{s\cdot P[j\,|\,C]}{|d|+s}$$

μ_j from corpus

with MLEs P[j|d], P[j|C]

with $\lambda = \frac{|d|}{|d|+s}$

where μ_1=sP[1|C], ..., μ_m=sP[m|C] are the parameters of the underlying Dirichlet distribution, with constant s > 1 typically set to multiple of document length

derived by MAP with Dirichlet distribution as prior for parameters of multinomial distribution

$$f(x_1,\ldots,x_m) = \Pi_{j=1..m} x_j^{\mu_j - 1} / B(\mu_1,\ldots,\mu_m) \quad \textbf{if} \quad \sum_{j=1..m} x_j = 1$$

with multinomial Beta: $B(\mu_1,\ldots,\mu_m) = \Pi_{j=1..m} \Gamma(\mu_j) \;/\; \Gamma(\Sigma_{j=1..m} \mu_j)$

(Dirichlet is conjugate prior for parameters of multinomial distribution: Dirichlet prior implies Dirichlet posterior, only with different parameters)

4.2.3 Extended LMs

<u>large variety of extensions:</u>

- **Term-specific smoothing**
  (JM with term-specific λ_j, e.g. based on idf values)
- **Parsimonious LM**
  (JM-style smoothing with smaller feature space)
- **N-gram (Sequence) Models (e.g. HMMs)**
- **(Semantic) Translation Models**
- **Cross-Lingual Models**
- **Query-Log- & Click-Stream-based LM**

(Semantic) Translation Model

$$P[q \mid d] = \prod_{j \in q} \sum_{w} P[j \mid w] \cdot P[w \mid d]$$

with word-word translation model P[j|w]

<u>Opportunities and difficulties:</u>

- **synonymy, hypernymy/hyponymy, polysemy**
- **efficiency**
- **training**

estimate P[j|w] by overlap statistics on background corpus (Dice coefficients, Jaccard coefficients, etc.)

Query-Log-Based LM (User LM)

<u>Idea:</u>

for current query q_k leverage
prior query history $H_q = q_1 \ldots q_{k-1}$ and
prior click stream $H_c = d_1 \ldots d_{k-1}$ as background LMs

<u>Example:</u>

q_k = „Java library" benefits from q_{k-1} = „cgi programming"

Simple Mixture Model with Fixed Coefficient Interpolation:

$$P[w \mid q_i] = \frac{freq(w, q_i)}{\mid q_i \mid} \qquad P[w \mid H_q] = \frac{1}{k-1} \sum_{i=1..k-1} P[w \mid q_i]$$

$$P[w \mid d_i] = \frac{freq(w, d_i)}{\mid d_i \mid} \qquad P[w \mid H_c] = \frac{1}{k-1} \sum_{i=1..k-1} P[w \mid d_i]$$

$$P[w \mid H_q, H_c] = \beta\, P[w \mid H_q] + (1-\beta)\, P[w \mid H_c]$$

$$P[w \mid \theta_k] = \alpha\, P[w \mid q_k] + (1-\alpha)\, P[w \mid H_q, H_c]$$

More advanced models with Dirichlet priors in the literature

Additional Literature for Chapter 4

Statistical Language Models:

- Grossman/Frieder Section 2.3
- W.B. Croft, J. Lafferty (Editors): Language Modeling for Information Retrieval, Kluwer, 2003
- C. Zhai: Statistical Language Models for Information Retrieval, Tutorial Slides,
- SIGIR 2005
- X. Liu, W.B. Croft: Statistical Language Modeling for Information Retrieval, Annual Review of Information Science and Technology 39, 2004
- J. Ponte, W.B. Croft: A Language Modeling Approach to Information Retrieval, SIGIR 1998
- C. Zhai, J. Lafferty: A Study of Smoothing Methods for Language Models Applied to Information Retrieval, TOIS 22(2), 2004
- C. Zhai, J. Lafferty: A Risk Minimization Framework for Information Retrieval, Information Processing and Management 42, 2006
- X. Shen, B. Tan, C. Zhai: Context-Sensitive Information Retrieval Using Implicit Feedback, SIGIR 2005
- M.E. Maron, J.L. Kuhns: On Relevance, Probabilistic Indexing, and Information Retrieval, Journal of the ACM 7, 1960