Chapter III:
Ranking Principles

Information Retrieval & Data Mining

Universität des Saarlandes, Saarbrücken

Winter Semester 2011/12

Chapter III: Ranking Principles*

III.1 Document Processing & Boolean Retrieval

Tokenization, Stemming, Lemmatization, Boolean Retrieval Models

III.2 Basic Ranking & Evaluation Measures

TF*IDF & Vector Space Model, Precision/Recall, F-Measure, MAP, etc.

III.3 Probabilistic Retrieval Models

Binary/Multivariate Models, 2-Poisson Model, BM25, Relevance Feedback

III.4 Statistical Language Models (LMs)

Basic LMs, Smoothing, Extended LMs, Cross-Lingual IR

III.5 Advanced Query Types

Query Expansion, Proximity Ranking, Fuzzy Retrieval, XML-IR

*mostly following **Manning/Raghavan/Schütze**, with additions from other sources

III.3 Probabilistic Information Retrieval

- III.3 Probabilistic IR *(MRS book, Chapter 11)*
  - 3.1 Multivariate Binary Model & Smoothing
  - 3.2 Poisson Model, Multinomial Model, Dirichlet Model
  - 3.3 Probabilistic IR with Poisson Model (Okapi BM25)
  - 3.4 Tree Dependence Model & Bayesian Nets for IR

TF*IDF vs. Probabilistic Models

- TF*IDF sufficiently effective in practice but often criticized for being "*too ad-hoc*"
- Typically outperformed by probabilistic ranking models and/or statistical language models in all of the **major IR benchmarks**:
  - TREC: http://trec.nist.gov/
  - CLEF: http://clef2011.org/
  - INEX: https://inex.mmci.uni-saarland.de/

- Family of Probabilistic IR Models
  - Generative models for documents as *bags-of-words*
  - Binary independence model vs. multinomial (& multivariate) models
- Family of Statistical Language Models
  - Generative models for documents (and queries) as *entire sequences of words*
  - Divergence of document and query distributions (e.g., Kullback-Leibler)

"Is This Document Relevant? … Probably"

A survey of probabilistic models in information retrieval.

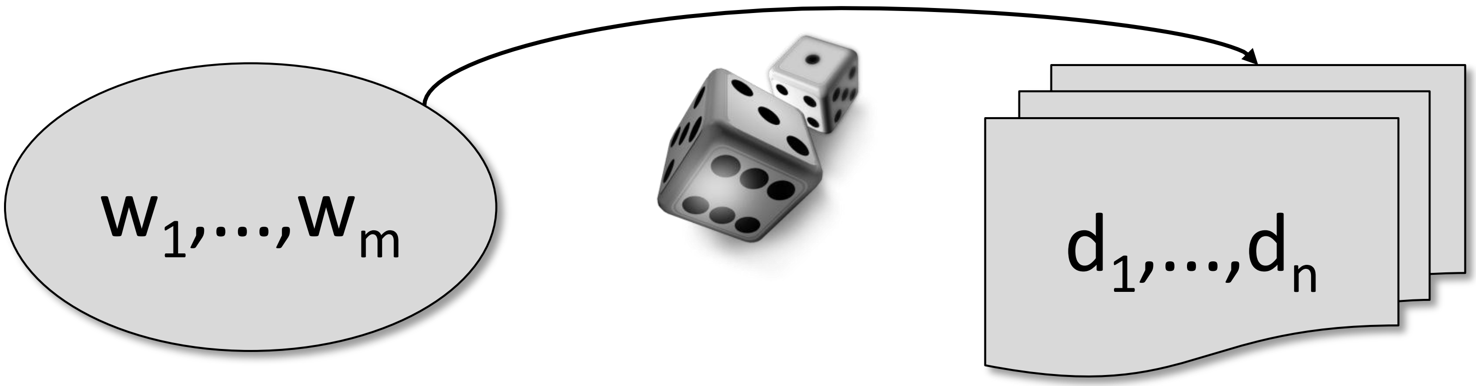
Fabio Crestani, Mounia Lalmas, Cornelis J. Van Rijsbergen, and Iain Campbell

Computer Science Department

University of Glasgow

Probabilistic IR

Based on **generative model**:
- probabilistic mechanism for producing document (or query)
- usually with specific family of parameterized distribution



Very powerful model but restricted through practical limitations:
- often with strong independence assumptions among words
- justified by "**curse of dimensionality**":

  corpus with n docs and m terms has $n = 2^m$ distinct possible docs
  would have to estimate model parameters from $n << 2^m$ docs
  (problems of sparseness & computational tractability)

III.3.1 Multivariate Binary Model

For generating doc d from joint (multivariate) word distribution ϕ
- consider *binary RVs*: $X_w = 1$ if word w occurs in doc d, 0 otherwise
- postulate *independence* among these RVs

$$P[d \mid \phi] = \prod_{w \in W} \phi_w^{\,X_w} \; (1-\phi_w)^{1-X_w}$$

with vocabulary W
and parameters (priors)

$$= \prod_{w \in D} \phi_w \prod_{w \in W, w \notin D} (1-\phi_w)$$

ϕ_w =
P[randomly drawn word is w]

<u>However:</u>
- presence of short documents underestimated
- product for absent words underestimates prob. of likely docs
- too much prob. mass given to very unlikely word combinations

Probabilistic Retrieval with the Binary Model

[Robertson and Sparck-Jones 1976]

Binary Relevance Model:

- Document d *is relevant* for query q (i.e., $R=1$) *or not* (i.e., $R=0$)
- Ranking based on *sim(doc d, query q) =*
  P[R=1|d,q] = P [doc d is relevant for query q |
  d has term vector $X_1,\ldots,X_m$ *]*

Probability Ranking Principle (PRP)

PRP with Costs: [Robertson 1977]

For a given retrieval task, the cost of retrieving d as the next result in a ranked list for query q is:

$cost(d,q) := C_1 * P[R=1|d,q] + C_0 * P[R=0|d,q]$ ("1/0 loss case")

with cost constants

C_1 = *cost of retrieving a relevant doc*

C_0 = *cost of retrieving an irrelevant doc*

For $C_1 < C_0$, the cost is minimized by choosing

$arg\ max_d\ P[R=1|d,q]$

Optimality of PRP

Goal:

Return top-k documents

in descending order of $P[R=1|d,q]$ or $cost(d,q)$, respectively.

Bayes' Optimal Decision Rule: (PRP without cost function)

Return documents which are more likely

to be relevant than irrelevant, i.e.:

Document d is relevant for query q

iff $P[R=1|d,q] > P[R=0|d,q]$

Theorem:

The PRP is optimal, in the sense that it minimizes the expected loss (aka. "Bayes' risk") under the 1/0 loss function.

Derivation of PRP

Consider doc d to be retrieved next,
i.e., d is preferred over all other candidate docs d'

cost(d) :=

$$C_1\, P[R=1|d] + C_0\, P[R=0|d] \;\le\; C_1\, P[R=1|d'] + C_0\, P[R=0|d']$$

$$=: \text{cost}(d')$$

$$\Leftrightarrow\; C_1\, P[R=1|d] + C_0\,(1 - P[R=1|d]) \le C_1\, P[R=1|d'] + C_0\,(1 - P[R=1|d'])$$

$$\Leftrightarrow\; C_1\, P[R=1|d] - C_0\, P[R=1|d] \;\le\; C_1\, P[R=1|d'] - C_0\, P[R=1|d']$$

$$\Leftrightarrow\; (C_1 - C_0)\, P[R=1|d] \;\le\; (C_1 - C_0)\, P[R=1|d']$$

$$\Leftrightarrow\; P[R=1|d] \;\ge\; P[R=1|d']$$

as $C_1 < C_0$ by assumption

for all d'

Binary Model and Independence

<u>Basic Assumption:</u>

Relevant and irrelevant documents differ in their term distribution.

Binary Independence Model (BIM) Model:

- Probabilities for term occurrences are *pairwise independent* for different terms.
- Term weights are *binary* $\in \{0,1\}$.

→ For terms that do not occur in query q, the probabilities of such a term to occur are the <u>same</u> among relevant and irrelevant documents.

→ Relevance of each document is <u>independent</u> of the relevance of any other document.

Ranking Proportional to Relevance Odds

$$sim(d,q) = O(R \mid d) = \frac{P[R=1 \mid d]}{P[R=0 \mid d]}$$ (using odds for relevance)

$$= \frac{P[d \mid R=1] \times P[R=1]}{P[d \mid R=0] \times P[R=0]}$$ (Bayes' theorem)

$$\propto \frac{P[d \mid R=1]}{P[d \mid R=0]} = \prod_{i=1}^{m} \frac{P[d_i \mid R=1]}{P[d_i \mid R=0]}$$ (independence or linked dependence)

$$= \prod_{i \in q} \frac{P[d_i \mid R=1]}{P[d_i \mid R=0]}$$ ($P[d_i|R=1] = P[d_i|R=0]$ for $i \notin q$)

$$= \prod_{\substack{i \in d \\ i \in q}} \frac{P[X_i = 1 \mid R=1]}{P[X_i = 1 \mid R=0]} \cdot \prod_{\substack{i \notin d \\ i \in q}} \frac{P[X_i = 0 \mid R=1]}{P[X_i = 0 \mid R=0]}$$

d_i = 1 if d includes term i, 0 otherwise

X_i = 1 if random doc includes term i, 0 otherwise

Ranking Proportional to Relevance Odds

$$= \prod_{\substack{i \in d \\ i \in q}} \frac{p_i}{q_i} \cdot \prod_{\substack{i \notin d \\ i \in q}} \frac{1-p_i}{1-q_i}$$

with *estimators* $p_i = P[X_i=1|R=1]$ and $q_i = P[X_i=1|R=0]$

$$= \prod_{i \in q} \frac{p_i^{d_i}}{q_i^{d_i}} \cdot \prod_{i \in q} \frac{(1-p_i)^{1-d_i}}{(1-q_i)^{1-d_i}}$$

with $d_i = 1$ iff $i \in d$, 0 otherwise

$$\propto \sum_{i \in q} \log\left(\frac{p_i^{d_i}(1-p_i)}{(1-p_i)^{d_i}}\right) - \log\left(\frac{q_i^{d_i}(1-q_i)}{(1-q_i)^{d_i}}\right)$$

$$= \sum_{i \in q} d_i \log \frac{p_i}{1-p_i} + \sum_{i \in q} d_i \log \frac{1-q_i}{q_i} + \sum_{i \in q} \log \frac{1-p_i}{1-q_i}$$

invariant of document d

$$\propto \sum_{i \in q} d_i \log \frac{p_i}{1-p_i} + \sum_{i \in q} d_i \log \frac{1-q_i}{q_i} \propto sim(d,q)$$

Probabilistic Retrieval:
Robertson/Sparck-Jones Formula

Estimate p_i und q_i based on *training sample* (query q on small sample of corpus) or based on intellectual assessment of first round's results (*relevance feedback*):

Let N be #docs in sample
R be # relevant docs in sample
n_i be #docs in sample that contain term i
r_i be #relevant docs in sample that contain term i

$\Rightarrow$ Estimate: $p_i = \frac{r_i}{R}$ $\quad q_i = \frac{n_i - r_i}{N - R}$

or: $p_i = \frac{r_i + 0.5}{R + 1}$ $\quad q_i = \frac{n_i - r_i + 0.5}{N - R + 1}$ (Lidstone smoothing with λ=0.5)

$$\Rightarrow \quad sim(d,q) = \sum_{i \in q} d_i \log \frac{r_i + 0.5}{R - r_i + 0.5} + \sum_{i \in q} d_i \log \frac{N - n_i - R + r_i + 0.5}{n_i - r_i + 0.5}$$

$\Rightarrow$ Weight of term i in doc d: $\log \frac{(r_i + 0.5)\,(N - n_i - R + r_i + 0.5)}{(R - r_i + 0.5)(n_i - r_i + 0.5)}$

Example for Probabilistic Retrieval

Documents $d_1 \ldots d_4$ with relevance feedback:

	t_1	t_2	t_3	t_4	t_5	t_6	**R**
d_1	1	0	1	1	0	0	**1**
d_2	1	1	0	1	1	0	**1**
d_3	0	0	0	1	1	0	**0**
d_4	0	0	1	0	0	0	**0**
n_i	2	1	2	3	2	0	
r_i	2	1	1	2	1	0	
p_i	5/6	1/2	1/2	5/6	1/2	1/6	
q_i	1/6	1/6	1/2	1/2	1/2	1/6	

q: t_1 t_2 t_3 t_4 t_5 t_6

N=4, R=2

Score of *new document* d_5 (smoothing omitted):

$d_5 \cap q$: <1 1 0 0 0 1> $\rightarrow$ sim(d_5, q) = log 5 + log 1 + log 1/5 + log 5 + log 5 + log 5

$$\text{using } sim(d,q) = \sum_{i \in q} d_i \log \frac{p_i}{1-p_i} + \sum_{i \in q} d_i \log \frac{1-q_i}{q_i}$$

Relationship to TF*IDF Formula

Assumptions (without training sample or relevance feedback):

- p_i is the same for all i
- most documents are irrelevant
- each individual term i is infrequent

This implies:

- $\sum_{i \in q} d_i \log \frac{p_i}{1-p_i} = c \sum_{i \in q} d_i$ *with constant c*
- $q_i = P[X_i = 1 | R = 0] \approx \frac{df_i}{N}$
- $\frac{1-q_i}{q_i} = \frac{N - df_i}{df_i} \approx \frac{N}{df_i}$

$$\Rightarrow \quad sim(d,q) = \sum_{i \in q} d_i \log \frac{p_i}{1-p_i} + \sum_{i \in q} d_i \log \frac{1-q_i}{q_i}$$

$$\approx c \sum_{i \in q} d_i + \sum_{i \in q} d_i \cdot \log idf_i$$

~ scalar product over the product of tf and dampend idf values for query terms

Laplace Smoothing (with Uniform Prior)

Probabilities p_i and q_i for term i are estimated
by **MLE for Binomial distribution**
(repeated coin tosses for relevant docs, showing term i with prob. p_i,
repeated coin tosses for irrelevant docs, showing term i with prob. q_i)

To avoid overfitting to feedback/training,
the estimates should be smoothed
(e.g., with uniform prior):

Instead of estimating $p_i = k/n$ estimate:

$p_i = (k+1) / (n+2)$ (Laplace's law of succession)

or with heuristic generalization:

$p_i = (k+\lambda) / (n+2\lambda)$ with $\lambda > 0$

(e.g., using λ=0.5) (Lidstone's law of succession)

And for Multinomial distribution (n times w-faceted dice) estimate:

$p_i = (k_i + 1) / (n + w)$

III.3.2 Advanced Models: Poisson/Multinomial

For generating doc d

- consider *counting RVs*: x_w = number of occurrences of w in d
- still postulate *independence* among these RVs

Poisson model with word-specific parameters μ_w:

$$P[d \mid \mu] = \prod_{w \in W} \frac{e^{-\mu_w} \cdot \mu_w^{\ x_w}}{x_w!} = e^{-\sum_{w \in W} \mu_w} \prod_{w \in d} \frac{\mu_w^{\ x_w}}{x_w!}$$

MLE for μ_w is straightforward <u>but</u>:

- no likelihood penalty by absent words
- no control of doc length

$$MLE\ \hat{\mu}_w = \frac{1}{n} \sum_{i=1}^{n} k_w$$

for n iid. samples (docs) with values k_w (word frequencies)

Multinomial Model

For generating doc d

- consider *counting RVs*: x_w = number of occurrences of w in d
- first generate doc length (a RV): $\ell_d = \Sigma_w x_w$
- then generate word frequencies x_w

$$P[\ell_d, \{x_w\} \mid \{\theta_w\}] = P[\ell_d] \cdot P[\{x_w\} \mid \ell_d, \{\theta_w\}]$$

$$= P[\ell_d] \cdot \binom{\ell_d}{\{x_w\}} \prod_{w \in W} \theta_w^{x_w}$$

$$= P[\ell_d] \cdot \ell_d! \prod_{w \in d} \frac{\theta_w^{x_w}}{x_w!}$$

with word-specific parameters θ_w = P[randomly drawn word is w]

Burstiness and the Dirichlet Model

Problem:

- In practice, words in documents do not appear independently
- Poisson/Multinomial underestimate likelihood of docs with high tf
- "bursty" word occurrences are not unlikely:
  - term may be frequent in doc but infrequent in corpus
  - for example, P[tf > 10] is low, but P[tf > 10 | tf > 0] is high

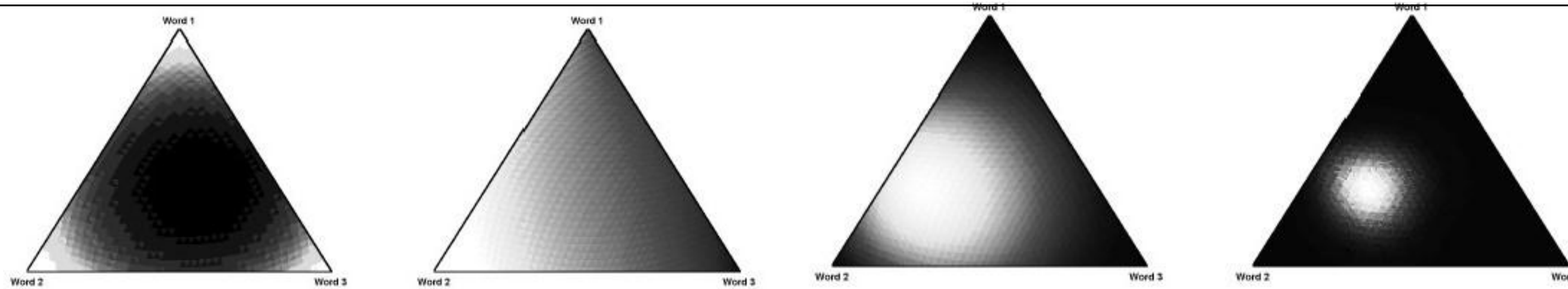
Solution: Two-level model

- **Hypergenerator**:
  to generate doc, first generate *word distribution in corpus*
  (thus obtain parameters of doc-specific generative model)
- **Generator**:
  then generate *word frequencies in doc*, using doc-specific model

Dirichlet Distribution as Hypergenerator for Two-Level Multinomial Model

$$P[\theta \mid \alpha] = \frac{\Gamma(\sum_w \alpha_w)}{\prod_w \Gamma(\alpha_w)} \prod_w \theta_w^{\alpha_w - 1} \quad \text{with} \quad \Gamma(x) = \int_0^\infty z^{x-1} e^{-z} dz$$

where $\Sigma_w \theta_w = 1$ and $\theta_w \geq 0$ and $\alpha_w \geq 0$ for all w



$\alpha = (0.44, 0.25, 0.31)$ (a) low-scaled DCM

$\alpha = (1.32, 0.75, 0.93)$ (b) mid-scaled DCM

$\alpha = (3.94, 2.25, 2.81)$ (c) high-scaled DCM

$\theta = (0.44, 0.25, 0.31)$ (d) multinomial

3-dimensional examples of Dirichlet and Multinomial
(Source: R.E. Madsen et al.: Modeling Word Burstiness Using the Dirichlet Distribution)

MAP of Multinomial with Dirichlet prior
is again Dirichlet (with different parameter values)
("Dirichlet is the conjugate prior of Multinomial")

MLE for Dirichlet Hypergenerator

$$P[d \mid \alpha] = \int_\theta P[\theta \mid \alpha] P[d \mid \theta] \, d\theta$$

2-step probability of generating doc d

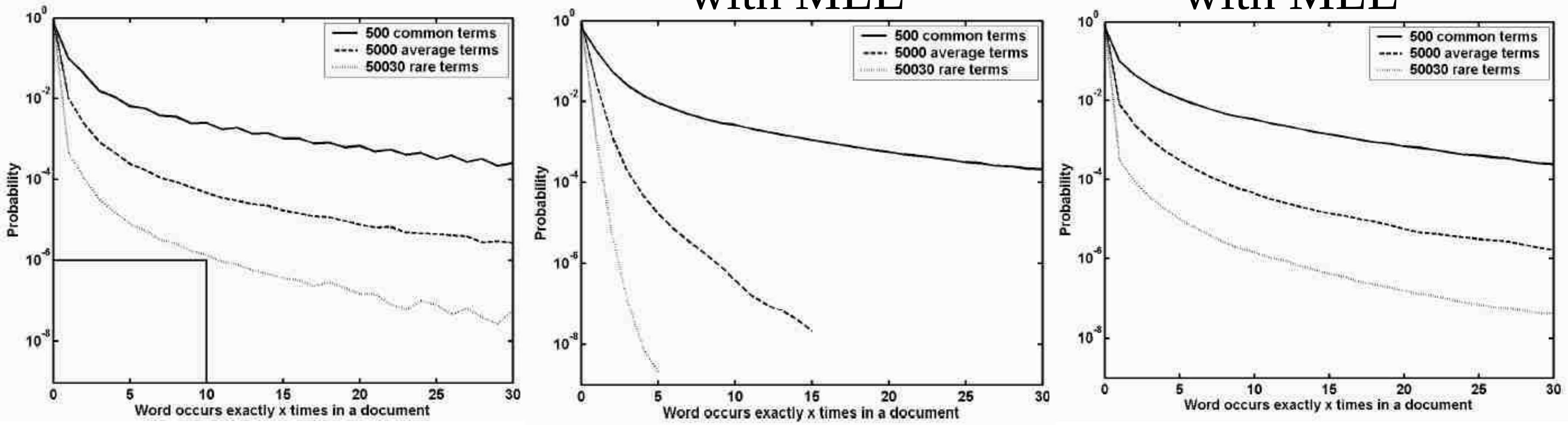
with independence assumptions:

$$P[d \mid \alpha] = P[\ell_d] \binom{\ell_d}{\{x_w\}} \frac{\Gamma(\sum_w \alpha_w)}{\Gamma(\sum_w (x_w + \alpha_w))} \prod_w \frac{\Gamma(x_w + \alpha_w)}{\Gamma(\alpha_w)}$$

for further steps for MLE use approximations and numerical methods (e.g., EM or Newton iterations)

Practical Adequacy of the Dirichlet Model

real corpus | Multinomial with MLE | Dirichlet with MLE



Source: R. Madsen et al.: Modeling Word Burstiness Using the Dirichlet Distribution, ICML 2005

model goodness for data $x_1, \ldots, x_n$ also measured by

perplexity $= 2^{-\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)}$ or $2^{-\sum_{i=1}^{n} freq(x_i) \log_2 p(x_i)}$

(i.e., the exponential of **entropy** or **cross-entropy**)

III.3.3 Probabilistic IR with Okapi BM25

Generalize term weight $w = \log \frac{p(1-q)}{q(1-p)}$

into $w = \log \frac{p_{tf}\, q_0}{q_{tf}\, p_0}$

with p_j, q_j denoting prob. that term occurs j times in rel./irrel. doc, resp.

Postulate Poisson (or 2-Poisson-mixture) distributions for terms:

$$p_{tf} = e^{-\lambda} \frac{\lambda^{tf}}{tf!} \qquad q_{tf} = e^{-\mu} \frac{\mu^{tf}}{tf!}$$

But: aim to reduce the number of parameters μ, λ that need to be learned from training samples!
Want: ad-hoc ranking function of similar ranking quality without training data!

Okapi BM25

Approximation of Poisson model by similarly-shaped function:

$$w := \log \frac{p(1-q)}{q(1-p)} \cdot \frac{tf}{k_1 + tf}$$

Finally leads to Okapi BM25 (with top-ranked results in TREC):

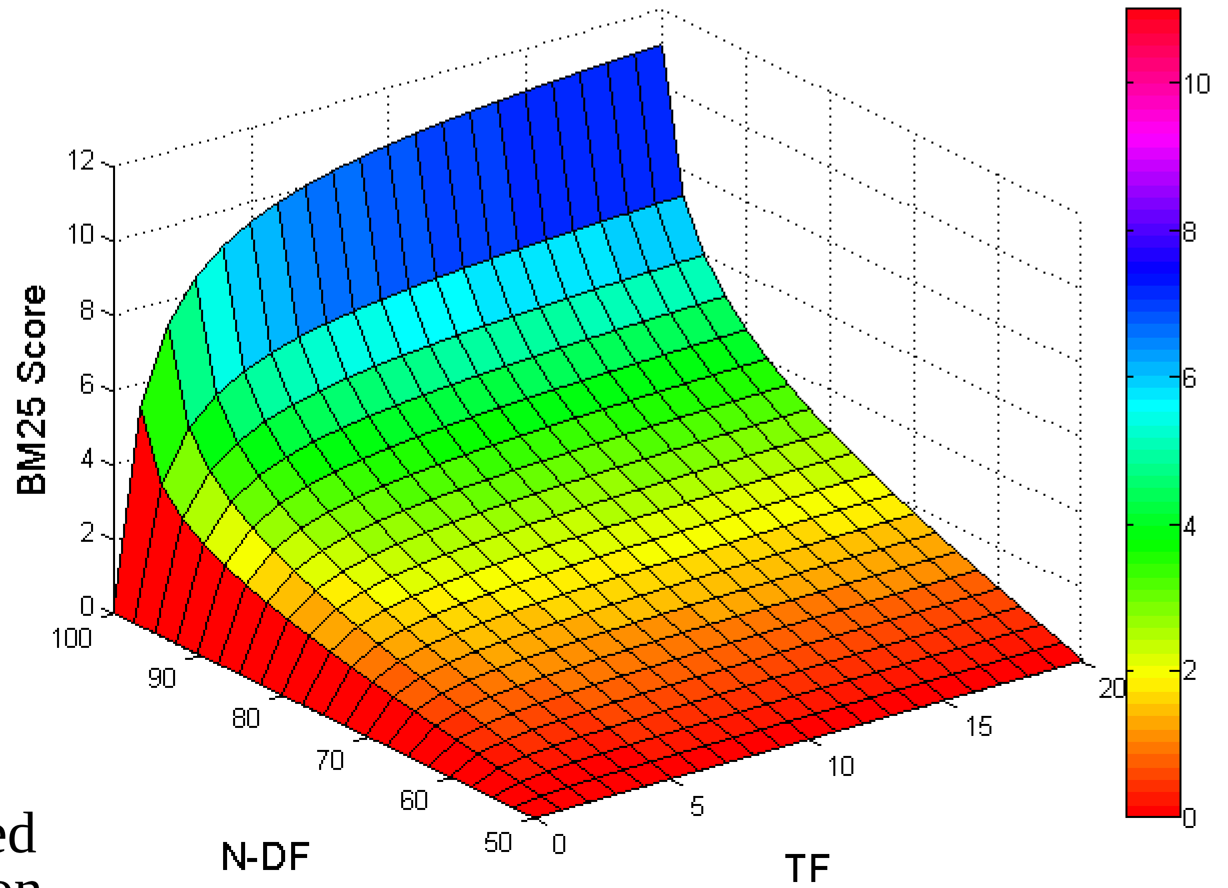
$$w_j(d) := \frac{(k_1+1)tf_j}{k_1((1-b) + b\frac{length(d)}{avg.doclength}) + tf_j} \cdot \log \frac{N - df_j + 0.5}{df_j + 0.5}$$

Or in its most comprehensive, tunable form: *score(d,q) :=*

$$\sum_{j=1..|q|} \log \frac{N - df_j + 0.5}{df_j + 0.5} \cdot \frac{(k_1+1)tf_j}{k_1((1-b) + b\frac{len(d)}{\Delta}) + tf_j} \cdot \frac{(k_3+1)qtf_j}{k_3 + qtf_j} + k_2 |q| \frac{\Delta - len(d)}{\Delta + len(d)}$$

with Δ=*avg.doclength*, tuning parameters k_1, k_2, k_3, b,
non-linear influence of tf, and consideration of *current doc length*

BM25 Example



- 3-d plot of a simplified BM25 scoring function using k_1=1.2 as parameter (DF is mirrored for better readability)
- scores for df>N/2 are negative!

$$w_j := \frac{(k_1+1)tf_j}{k_1+tf_j} \cdot \log \frac{N-df_j+0.5}{df_j+0.5}$$

III.3.4 Extensions to Probabilistic IR

Consider term correlations in documents (with binary X_i)
$\rightarrow$ Problem of estimating m-dimensional prob. distribution
$P[X_1 = \ldots \wedge X_2 = \ldots \wedge \ldots \wedge X_m = \ldots] =: f_X(X_1, \ldots, X_m)$

One possible approach: **Tree Dependence Model**

a) Consider only 2-dimensional probabilities (for term pairs i,j)

$$f_{ij}(X_i, X_j) = P[X_i = .. \wedge X_j = ..] = \sum_{X_1} .. \sum_{X_{i-1}} \sum_{X_{i+1}} .. \sum_{X_{j-1}} \sum_{X_{j+1}} .. \sum_{X_m} P[X_1 = \ldots \wedge .. \wedge X_m = \ldots]$$

b) For each term pair i,j

estimate the error between independence and the actual correlation

c) Construct a tree with terms as nodes and the

m-1 highest error (or correlation) values as weighted edges

Considering Two-dimensional Term Correlations

Variant 1:

Error of approximating f by g (**Kullback-Leibler divergence**) with g assuming pairwise term independence:

$$\varepsilon(f,g) := \sum_{\vec{X} \in \{0,1\}^m} f(\vec{X}) \log \frac{f(\vec{X})}{g(\vec{X})} = \sum_{\vec{X} \in \{0,1\}^m} f(\vec{X}) \log \frac{f(\vec{X})}{\prod_{i=1}^{m} g_i(X_i)}$$

Variant 2:

Correlation coefficient for term pairs:

$$\rho(X_i, X_j) := \frac{Cov(X_i, X_j)}{\sqrt{Var(X_i)}\sqrt{Var(X_j)}}$$

Variant 3:

level-α values or p-values

of **Chi-square independence test**

Example for Approximation Error ε (KL Strength)

<u>m=2:</u>

given are documents:

$d_1=(1,1)$, $d_2(0,0)$, $d_3=(1,1)$, $d_4=(0,1)$

estimation of 2-dimensional prob. distribution f:

$f(1,1) = P[X_1=1 \wedge X_2=1] = 2/4$

$f(0,0) = 1/4$, $f(0,1) = 1/4$, $f(1,0) = 0$

estimation of 1-dimensional marginal distributions g_1 and g_2:

$g_1(1) = P[X_1=1] = 2/4$, $g_1(0) = 2/4$

$g_2(1) = P[X_2=1] = 3/4$, $g_2(0) = 1/4$

estimation of 2-dim. distribution g with independent X_i:

$g(1,1) = g_1(1)*g_2(1) = 3/8$,

$g(0,0) = 1/8$, $g(0,1) = 3/8$, $g(1,0) =1/8$

approximation error ε (KL divergence):

$\varepsilon = 2/4 \log 4/3 \;+\; 1/4 \log 2 \;+\; 1/4 \log 2/3 \;+ 0$

Constructing the Term Dependence Tree

<u>Given:</u>

Complete graph (V, E) with m nodes $X_i \in V$ and
m^2 undirected edges $\in$ E with weights ε (or ρ)

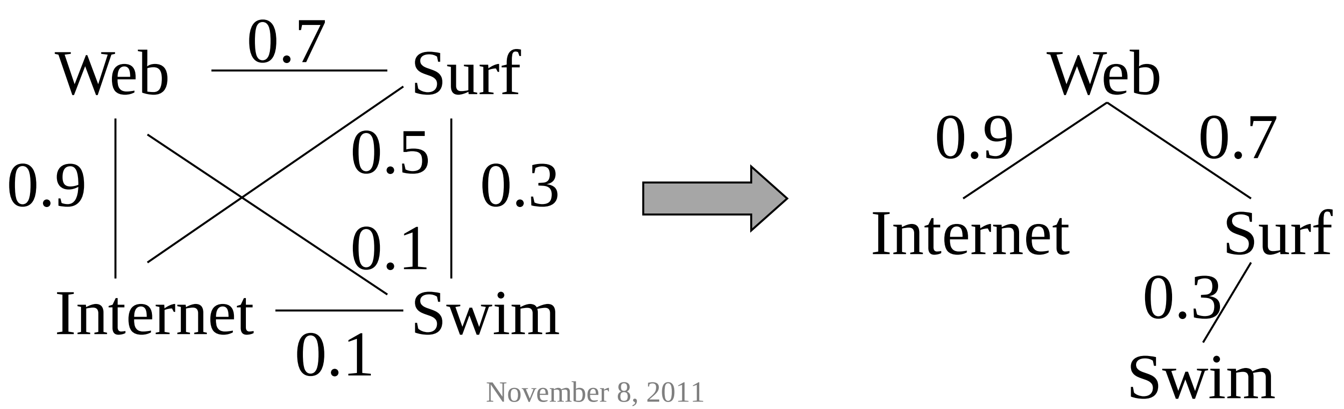
<u>Wanted:</u>

Spanning tree (V, E') with maximal sum of weights

<u>Algorithm:</u>

Sort the m^2 edges of E in descending order of weights
E' := $\emptyset$
Repeat until |E'| = m-1
  E' := E' $\cup$ {(i,j) $\in$ E | (i,j) has max. weight in E}
  provided that E' remains acyclic;
  E := E – {(i,j) $\in$ E | (i,j) has max. weight in E}

<u>Example:</u>



Estimation of Multidimensional Probabilities with Term Dependence Tree

Given is a term dependence tree (V = {X_1, ..., X_m}, E').
Let X_1 be the root, nodes are preorder-numbered, and assume that X_i and X_j are independent for (i,j) $\notin$ E'. Then:

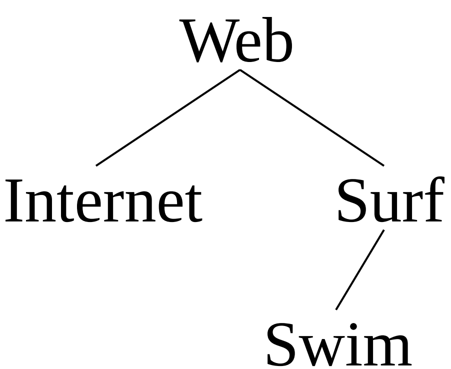
$$P[X_1 = .. \wedge .. \wedge X_m = ..] = P[X_1 = ..] P[X_2 = .. \wedge X_m = .. | X_1 = ..] \quad \text{cond. prob.}$$

$$= \Pi_{i=1..m} P[X_i = .. | X_1 = .. \wedge X_{i-1} = ..] \quad \text{chain rule}$$

$$= P[X_1] \cdot \prod_{(i,j) \in E'} P[X_j | X_i] \quad \text{cond. indep.}$$

$$= P[X_1] \cdot \prod_{(i,j) \in E'} \frac{P[X_i, X_j]}{P[X_i]} \quad \text{cond. prob.}$$

Example:



$$P[Web, Internet, Surf, Swim] =$$

$$P[Web] \frac{P[Web, Internet]}{P[Web]} \frac{P[Web, Surf]}{P[Web]} \frac{P[Surf, Swim]}{P[Surf]}$$

Bayesian Networks

A **Bayesian network (BN) is a directed, acyclic graph (V, E)** with the following properties:

- Nodes $\in$ V representing random variables and
- Edges $\in$ E representing dependencies.
- For a root R $\in$ V the BN captures the prior probability P[R = ...].
- For a node X $\in$ V with parents *parents(X) = {P_1, ..., P_k}* the BN captures the conditional probability $P[X=\ldots \mid P_1, \ldots, P_k]$.
- Node X is conditionally independent of a non-parent node Y given its parents *parents(X) = {P_1, ..., P_k}*: $P[X \mid P_1, \ldots, P_k, Y] = P[X \mid P_1, \ldots, P_k]$.

<u>This implies</u>:

$$P[X_1 \ldots X_n] = P[X_1 \mid X_2 \ldots X_n] \, P[X_2 \ldots X_n]$$

- by the chain rule:

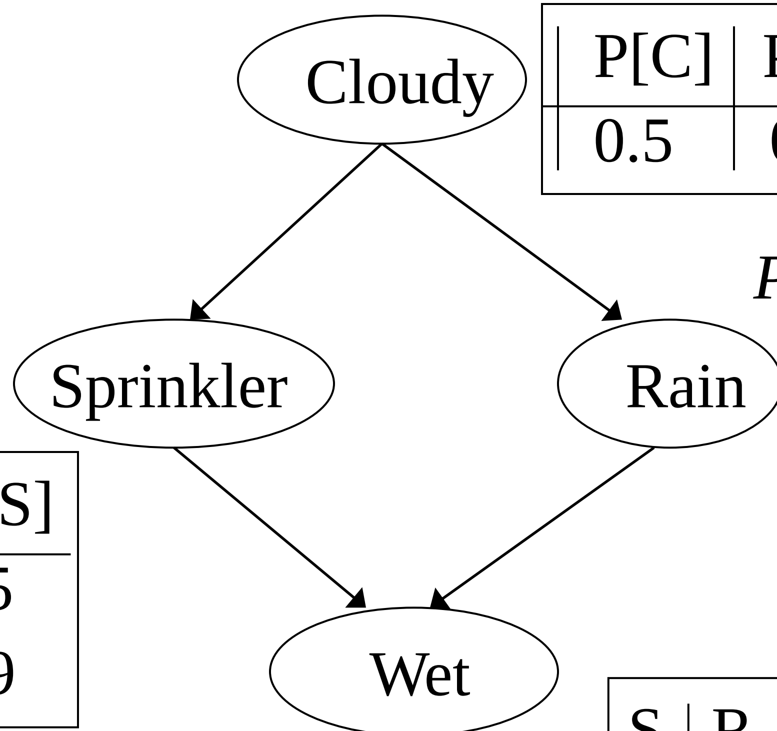
$$= \prod_{i=1}^{n} P[X_i \mid X_{(i+1)} \ldots X_n]$$

- by cond. independence:

$$= \prod_{i=1}^{n} P[X_i \mid parents(X_i), other\ nodes]$$

$$= \prod_{i=1}^{n} P[X_i \mid parents(X_i)]$$

Example of Bayesian Network
(aka. “Belief Network”)



P[C]:

P[C]	P[¬C]
0.5	0.5

P[S | C]:

C	P[S]	P[¬S]
F	0.5	0.5
T	0.1	0.9

P[R | C]:

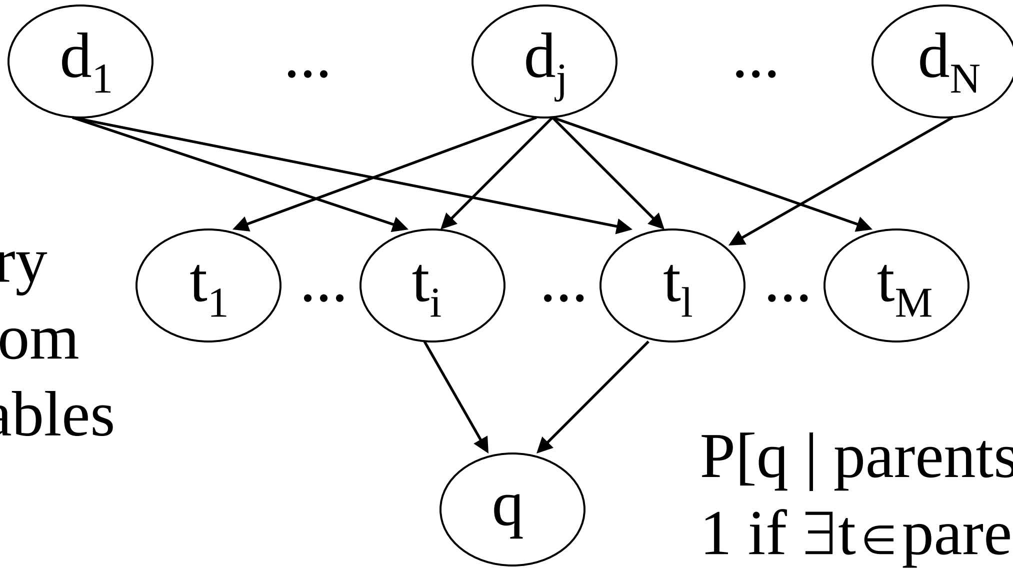
C	P[R]	P[¬R]
F	0.2	0.8
T	0.8	0.2

P[W | S,R]:

S	R	P[W]	P[¬W]
F	F	0.0	1.0
F	T	0.9	0.1
T	F	0.9	0.1
T	T	0.99	0.01

Bayesian Inference Networks for IR

with binary random variables



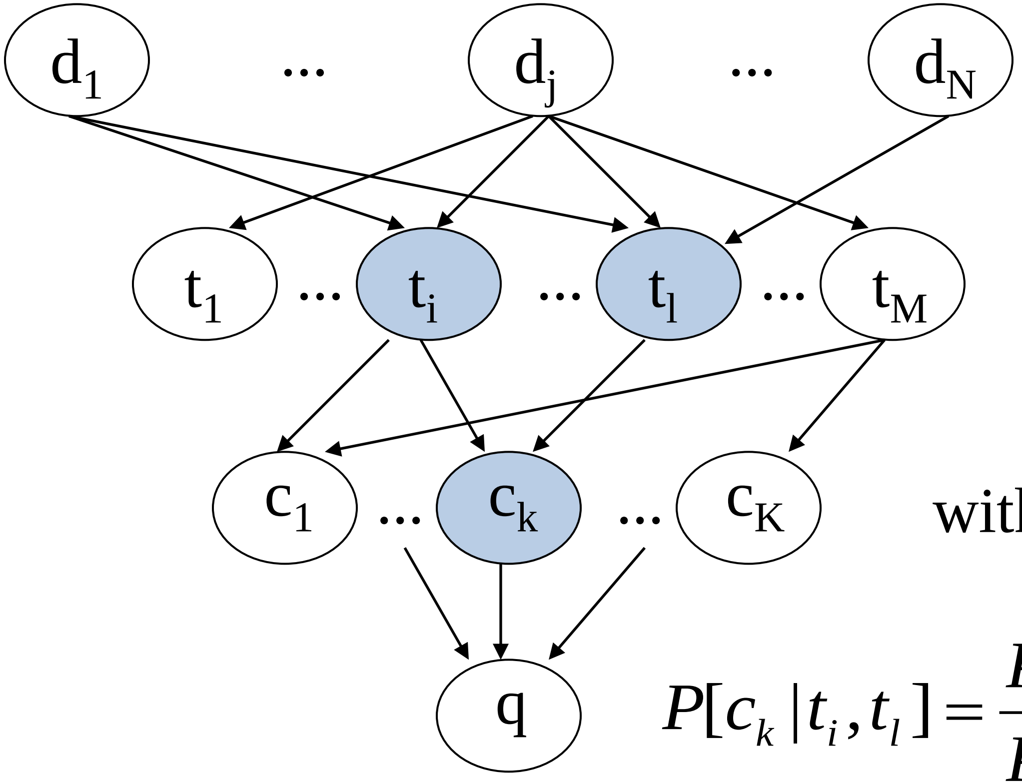
$P[d_j]=1/N$

$P[t_i \mid d_j \in parents(t_i)] =$
1 if t_i occurs in d_j,
0 otherwise

$P[q \mid parents(q)] =$
1 if $\exists t \in parents(q)$: t is relevant for q,
0 otherwise

$$
\begin{aligned}
P[q \wedge d_j] &= \sum_{(t_1 \ldots t_M)} P[q \wedge d_j \mid t_1 \ldots t_M]\, P[t_1 \ldots t_M] \\
&= \sum_{(t_1 \ldots t_M)} P[q \wedge d_j \wedge t_1 \wedge \ldots \wedge t_M] \\
&= \sum_{(t_1 \ldots t_M)} P[q \mid d_j \wedge t_1 \wedge \ldots \wedge t_M]\, P[d_j \wedge t_1 \wedge \ldots \wedge t_M] \\
&= \sum_{(t_1 \ldots t_M)} P[q \mid t_1 \wedge \ldots \wedge t_M]\, P[t_1 \wedge \ldots \wedge t_M \mid d_j]\, P[d_j]
\end{aligned}
$$

Advanced Bayesian Network for IR



with concepts / topics c_k

$$P[c_k \mid t_i, t_l] = \frac{P[t_i \wedge t_l]}{P[t_i \vee t_l]} \approx \frac{df_{il}}{df_i + df_l - df_{il}}$$

<u>Problems:</u>

- parameter estimation (sampling / training)
- (non-) scalable representation
- (in-) efficient prediction
- fully convincing experiments

Summary of Section III.3

- **Probabilistic IR** reconciles principled foundations
  with practically effective ranking
- Parameter estimation requires **smoothing** to avoid **overfitting**
- **Poisson-model**-based **Okapi BM25** has won many benchmarks
- **Multinomial & Dirichlet models** are even more expressive
- Extensions with **term dependencies**, such as **Bayesian Networks**,
  are intractable for general-purpose IR but interesting for specific apps

Additional Literature for Section III.3

- Manning/Raghavan/Schuetze, Chapter 11
- K. van Rijsbergen: Information Retrieval, Chapter 6: Probabilistic Retrieval, 1979, http://www.dcs.gla.ac.uk/Keith/Preface.html
- R. Madsen, D. Kauchak, C. Elkan: Modeling Word Burstiness Using the Dirichlet Distribution, ICML 2005
- S.E. Robertson, K. Sparck Jones: Relevance Weighting of Search Terms, JASIS 27(3), 1976
- S.E. Robertson, S. Walker: Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval, SIGIR 1994
- A. Singhal: Modern Information Retrieval – a Brief Overview, IEEE CS Data Engineering Bulletin 24(4), 2001
- K.W. Church, W.A. Gale: Poisson Mixtures, Natural Language Engineering 1(2), 1995
- C.T. Yu, W. Meng: Principles of Database Query Processing for Advanced Applications, Morgan Kaufmann, 1997, Chapter 9
- D. Heckerman: A Tutorial on Learning with Bayesian Networks, Technical Report MSR-TR-95-06, Microsoft Research, 1995
- S. Chaudhuri, G. Das, V. Hristidis, G. Weikum: Probabilistic information retrieval approach for ranking of database query results, TODS 31(3), 2006.