

Probabilistic Information Retrieval

CE-324: Modern Information Retrieval

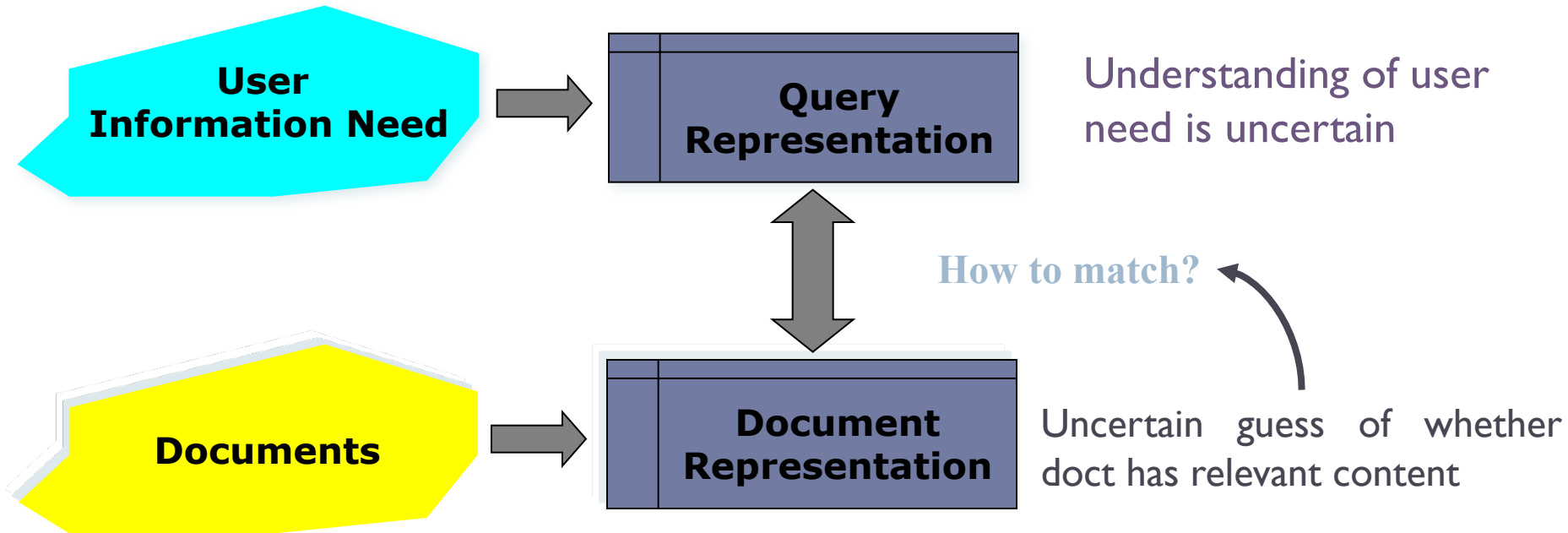
Sharif University of Technology

M. Soleymani

Spring 2020

Most slides have been adapted from: Profs. Manning, Nayak & Raghavan (CS-276, Stanford)

Why probabilities in IR?



In traditional IR systems, matching between each doc and query is attempted in a semantically imprecise space of index terms.

Probabilities provide a principled foundation for uncertain reasoning.

Can we use probabilities to quantify our uncertainties?

Probabilistic IR

- ▶ Probabilistic methods are one of the oldest but also one of the currently hottest topics in IR.
 - ▶ Traditionally: neat ideas, but didn't win on performance
 - ▶ It may be different now.

Probabilistic IR topics

- ▶ Classical probabilistic retrieval model
 - ▶ **Probability Ranking Principle**
 - ▶ Binary independence model (\approx We will see that its a Naïve Bayes text categorization)
 - ▶ (Okapi) BM25
- ▶ Language model approach to IR
 - ▶ An important emphasis on this approach in recent work

The document ranking problem

- ▶ Problem specification:
 - ▶ We have a collection of docs
 - ▶ User issues a query
 - ▶ A list of docs needs to be returned
- ▶ Ranking method is the core of an IR system:
 - ▶ In what order do we present documents to the user?
- ▶ Idea: Rank by probability of relevance of the doc w.r.t. information need
 - ▶ $P(R = 1 | doc_i, query)$

Probability Ranking Principle (PRP)

“If a reference retrieval system’s response to each request is a **ranking** of the docs in the collection **in order of decreasing probability of relevance** to the user who submitted the request, where the **probabilities are estimated as accurately as possible** on the basis of whatever data have been made available to the system for this purpose, the **overall effectiveness of the system to its user will be the best** that is obtainable on the basis of those data.”

[1960s/1970s] S. Robertson, W.S. Cooper, M.E. Maron; van Rijsbergen (1979:113); Manning & Schütze (1999:538)

Recall a few probability basics

► Product rule: $p(a, b) = p(a|b)p(b)$

► Sum rule: $p(a) = \sum_b p(a, b)$

► Bayes' Rule

$$\underset{\substack{\uparrow \\ \text{Posterior}}}{p(a|b)} = \frac{p(b|a) \overset{\substack{\nwarrow \\ \text{Prior}}}{p(a)}}{p(b)} = \frac{p(b|a)p(a)}{p(b|a)p(a) + p(b|\bar{a})p(\bar{a})}$$

► Odds:

$$O(a) = \frac{p(a)}{p(\bar{a})} = \frac{p(a)}{1 - p(a)}$$

Probability Ranking Principle (PRP)

d : doc

q : query

R : **relevance** of a doc w.r.t. given (fixed) query

$R = 1$: relevant

$R = 0$: not relevant

Need to find probability that a doc x is relevant to a query q .

$$p(R = 1|d, q)$$

$$p(R = 0|d, q) = 1 - p(R = 1|d, q)$$

Probability Ranking Principle (PRP)

$$p(R = 1|d, q) = \frac{p(d|R = 1, q)p(R = 1|q)}{p(d|q)}$$

$$p(R = 0|d, q) = \frac{p(d|R = 0, q)p(R = 0|q)}{p(d|q)}$$

$p(R=1|q), p(R=0|q)$ - prior probability of retrieving a relevant or non-relevant doc at random (for query q)

- ▶ $p(d|R = 1, q)$: probability of d in the class of relevant docs to the query q .
- ▶ $p(d|R = 0, q)$: probability of d in the class of non-relevant docs to the query q .

Probability Ranking Principle (PRP)

- ▶ How do we compute all those probabilities?
 - ▶ Do not know exact probabilities, have to use estimates
 - ▶ Binary Independence Model (BIM)
 - ▶ which we discuss next – is the simplest model

Probabilistic Retrieval Strategy

- ▶ Estimate how terms contribute to relevance
 - ▶ How do things like tf, df, and length influence your judgments about doc relevance?
 - ▶ A more nuanced answer is the Okapi formula
 - Spärck Jones / Robertson
- ▶ Combine the above estimated values to find doc relevance probability
- ▶ Order docs by decreasing probability

Probabilistic Ranking

Basic concept:

“For a given query, if we know some docs that are relevant, terms that occur in those docs should be given greater weighting in searching for other relevant docs.

By making assumptions about the distribution of terms and applying Bayes Theorem, it is possible to derive weights theoretically.”

Van Rijsbergen

Binary Independence Model

- ▶ Traditionally used in conjunction with PRP
- ▶ **“Binary” = Boolean**: docs are represented as binary incidence vectors of terms
 - ▶ $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_m]$
 - ▶ $x_i = 1$ iff term i is present in document x .
- ▶ **“Independence”**: terms occur in docs independently
- ▶ Equivalent to Multivariate Bernoulli Naive Bayes model
 - ▶ Sometimes used for text categorization [we will see in the next lectures]

Binary Independence Model

- ▶ Will use odds and Bayes' Rule:

$$O(R | q, \mathbf{x}) = \frac{P(R = 1 | q, \mathbf{x})}{P(R = 0 | q, \mathbf{x})} = \frac{\frac{P(R = 1 | q)P(\mathbf{x} | R = 1, q)}{P(\mathbf{x} | q)}}{\frac{P(R = 0 | q)P(\mathbf{x} | R = 0, q)}{P(\mathbf{x} | q)}}$$

Binary Independence Model

$$O(R | q, \mathbf{x}) = \frac{P(R = 1 | q, \mathbf{x})}{P(R = 0 | q, \mathbf{x})} = \underbrace{\frac{P(R = 1 | q)}{P(R = 0 | q)}}_{\text{Constant for a given query}} \cdot \underbrace{\frac{P(\mathbf{x} | R = 1, q)}{P(\mathbf{x} | R = 0, q)}}_{\text{Needs estimation}}$$

Using **Independence** Assumption:

$$\frac{p(\mathbf{x} | R = 1, q)}{p(\mathbf{x} | R = 0, q)} = \prod_{i=1}^n \frac{P(x_i | R = 1, q)}{P(x_i | R = 0, q)}$$

$$O(R | q, d) = O(R | q) \cdot \prod_{i=1}^n \frac{P(x_i | R = 1, q)}{P(x_i | R = 0, q)}$$

Binary Independence Model

Since x_i is either 0 or 1:

$$O(R | q, d) = O(R | q) \cdot \prod_{x_i=1} \frac{P(x_i = 1 | R = 1, q)}{P(x_i = 1 | R = 0, q)} \cdot \prod_{x_i=0} \frac{P(x_i = 0 | R = 1, q)}{P(x_i = 0 | R = 0, q)}$$

Let $p_i = P(x_i = 1 | R = 1, q)$

$$p_i = P(x_i = 1 | R = 1, q)$$

Assume, for all terms not occurring in the query ($q_i=0$) that $p_i = u_i$

This can be changed (e.g., in relevance feedback)

Probabilities

document	relevant ($R=1$)	not relevant ($R=0$)
term present $x_i = 1$	p_i	u_i
term absent $x_i = 0$	$(1 - p_i)$	$(1 - u_i)$

Then...

Binary Independence Model

$$O(R \mid q, \mathbf{x}^r) = \boxed{O(R \mid q)} \cdot \prod_{x_i = q_i = 1} \frac{p_i}{u_i} \cdot \prod_{\substack{x_i = 0 \\ q_i = 1}} \frac{1 - p_i}{1 - u_i}$$

All matching terms

Non-matching
query terms

$$= \boxed{O(R \mid q)} \cdot \prod_{x_i = q_i = 1} \frac{p_i (1 - u_i)}{u_i (1 - p_i)} \cdot \prod_{q_i = 1} \frac{1 - p_i}{1 - u_i}$$

All matching terms

All query terms

Binary Independence Model

$$O(R | q, \mathbf{x}^r) = \underbrace{O(R | q)}_{\text{Constant for each query}} \cdot \prod_{x_i=q_i=1} \frac{p_i (1-u_i)}{u_i (1-p_i)} \cdot \prod_{q_i=1} \frac{1-p_i}{1-u_i}$$

Only quantity to be estimated for rankings

Retrieval Status Value:

$$RSV = \log \prod_{x_i=q_i=1} \frac{p_i (1-u_i)}{u_i (1-p_i)} = \sum_{x_i=q_i=1} \log \frac{p_i (1-u_i)}{u_i (1-p_i)}$$

Binary Independence Model

All boils down to computing RSV:

$$RSV = \log \prod_{x_i=q_i=1} \frac{p_i(1-u_i)}{u_i(1-p_i)} = \sum_{x_i=q_i=1} \log \frac{p_i(1-u_i)}{u_i(1-p_i)}$$

$$RSV = \sum_{x_i=q_i=1} c_i; \quad c_i = \log \frac{p_i(1-u_i)}{u_i(1-p_i)}$$

c_i s function as the term weights in this model

c_i s are **log odds ratios**

So, how do we compute c_i 's from our data ?

BIM: example

- ▶ $q = \{x_1, x_2\}$
- ▶ Relevance judgements from 20 docs together with the distribution of x_1, x_2 within these docs

d_i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
x_1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
x_2	1	1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0
$r(q, d_i)$	R	R	R	R	\bar{R}	R	R	R	R	\bar{R}	\bar{R}	R	R	R	\bar{R}	\bar{R}	\bar{R}	R	\bar{R}	\bar{R}

- ▶ $p_1 = 8/12, u_1 = 3/8$
- ▶ $p_2 = 7/12$ and $u_2 = 4/8$.
- ▶ $c_1 = \log 10 / 3$
- ▶ $c_2 = \log 7 / 5$

$(1,1)$
 $(1,0)$
 $(0,1)$
 $(0,0)$

Binary Independence Model

Estimating RSV coefficients in theory

For each term i look at this table of document counts:


Documents	Relevant	Non-Relevant	Total
$x_i=1$	s	$df-s$	df
$x_i=0$	$S-s$	$N-df-S+s$	$N-df$
Total	S	$N-S$	N

Estimates: $p_i \approx \frac{s}{S}$ $u_i = \frac{df-s}{N-S}$

Weight of i -th term: $c_i \approx \log \frac{s/(S-s)}{(df-s)/(N-df-S+s)}$

For now,
assume no
zero terms.

Estimation – key challenge

- ▶ If non-relevant docs are approximated by the whole collection:
 - ▶ $u_i = df_i/N$
 - ▶ prob. of occurrence in non-relevant docs for query
 - ▶ $\log(1-u_i)/u_i = \log(N-df_i)/df_i \approx \log N/df_i$  **IDF!**

Estimation – key challenge

- ▶ p_i cannot be approximated as easily as u_i
 - ▶ probability of occurrence in relevant docs
- ▶ p_i can be estimated in various ways:
 - ▶ constant (Croft and Harper combination match)
 - ▶ Then just get idf weighting of terms ($p_i = 0.5, RSV = \sum_{q_i=x_i=1} \log \frac{N}{df_i}$)
 - ▶ proportional to prob. of occurrence in collection
 - ▶ Greiff (SIGIR 1998) argues for $1/3 + 2/3 df_i/N$
 - ▶ from relevant docs if know some
 - ▶ Relevance weighting can be used in a feedback loop

Probabilistic Relevance Feedback

1. Guess p_i and u_i and use it to retrieve a first set of relevant docs VR .
2. Interact with the user to refine the description: user specifies some definite members with $R = 1$ (the set VR) and $R = 0$ (the set VNR)
3. Re-estimate p_i and u_i :

$$p_i = \frac{|VR_i| + \frac{1}{2}}{|VR| + 1}, \quad u_i = \frac{|VNR_i| + \frac{1}{2}}{|VNR| + 1}$$

4. Repeat, thus generating a succession of approximations to relevant docs

Probabilistic Relevance Feedback

1. Guess p_i and u_i and use it to retrieve a first set of relevant docs VR .
2. Interact with the user to refine the description: learn some definite members with $R = 1$ and $R = 0$
3. Re-estimate p_i and u_i :
 - ▶ Or can combine new info with original guess (use Bayesian update):

$$p_i^{(t+1)} = \frac{|VR_i| + \kappa p_i^{(t)}}{|VR| + \kappa}$$

κ is prior weight

4. Repeat, thus generating a succession of approximations to relevant docs

Iteratively estimating p_i (= Pseudo-relevance feedback)

1. Assume that p_i is constant over all x_i in query
 - ▶ $p_i = 0.5$ (even odds) for any given doc
2. Determine guess of relevant doc set:
 - ▶ V is fixed size set of highest ranked docs on this model
3. We need to improve our guesses for p_i and u_i :
 - ▶ Let V_i be set of docs containing x_i

$$p_i = \frac{|V_i| + 1/2}{|V| + 1}$$

- ▶ Assume if not retrieved then not relevant

$$u_i = \frac{df_i - |V_i| + 1/2}{N - |V| + 1}$$

4. Go to 2. until converges then return ranking

PRP and BIM

- ▶ Getting reasonable approximations of probabilities is possible.
- ▶ Requires restrictive assumptions:
 - ▶ boolean representation of docs/queries/relevance
 - ▶ term independence
 - ▶ terms that do not appear in the query don't affect the outcome
 - ▶ doc relevance values are independent
- ▶ Some of these assumptions can be removed
- ▶ Problem: either require partial relevance information or only can derive somewhat inferior term weights

Removing term independence

- In general, index terms aren't independent
 - Dependencies can be complex
- Rijsbergen (1979) proposed model of simple tree dependencies
 - In 1970s, estimation problems held back success of this model
 - Exactly Friedman and Goldszmidt's Tree Augmented Naive Bayes (AAAI 13, 1996)
 - Each term dependent on one other

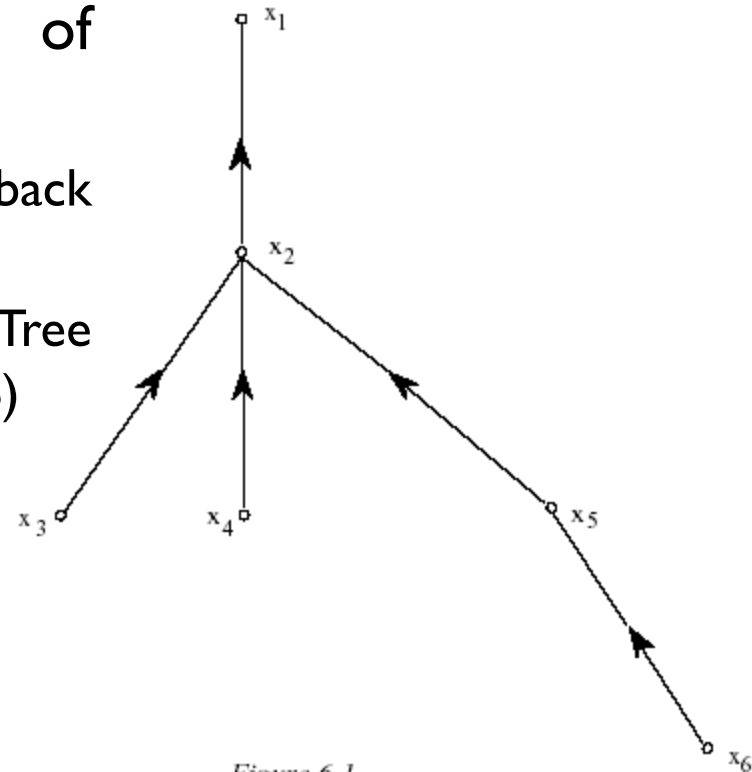


Figure 6.1.

A key limitations of the BIM

- ▶ BIM was designed for titles or abstracts, and not for modern full text search
 - ▶ like much of original IR
- ▶ We want to pay attention to term frequency and doc lengths
 - ▶ just like in other models we've discussed.

Okapi BM25

- ▶ BM25 “Best Match 25” (they had a bunch of tries!)
 - ▶ Developed in the context of the Okapi system
 - ▶ Started to be increasingly adopted by other teams during the TREC competitions
 - ▶ It works well
- ▶ Goal: Releasing some assumption of BIM while not adding too many parameters
 - ▶ (Spärck Jones et al. 2000)
- ▶ I’ll omit the theory, but show the form....

Recall: BIM

- Boils down to:

$$RSV^{BIM} = \sum_{x_i=q_i=1} c_i^{BIM}; \quad c_i^{BIM} = \log \frac{p_i(1-u_i)}{(1-p_i)u_i} \quad \leftarrow \text{Log odds ratio}$$

	document	relevant (R=1)	not relevant (R=0)
term present	$x_i = 1$	p_i	u_i
term absent	$x_i = 0$	$(1 - p_i)$	$(1 - u_i)$

- Simplifies to (with constant $p_i = 0.5$)

$$RSV^{BIM} = \sum_{x_i=q_i=1} \log \frac{N}{df_i}$$



“Early” versions of BM25

- ▶ Version 1: using the saturation function

$$c_i^{BM25v1}(tf_i) = c_i^{BIM} \frac{tf_i}{k_1 + tf_i}$$

- ▶ Version 2: BIM simplification to IDF:

$$c_i^{BM25v2}(tf_i) = \log \frac{N}{df_i} \times \frac{(k_1 + 1)tf_i}{k_1 + tf_i}$$

- ▶ $(k_1 + 1)$ factor doesn't change ranking, but makes term score 1 when $tf_i = 1$
- ▶ Similar to $tf-idf$, but term scores are bounded



Document length normalization

- ▶ Longer documents are likely to have larger tf_i values
- ▶ Why might documents be longer?
 - ▶ Verbosity: suggests observed tf_i too high
 - ▶ Larger scope: suggests observed tf_i may be right
- ▶ A real document collection probably has both effects
 - ▶ ... so should apply some kind of normalization



Document length normalization

- ▶ Document length:

$$dl = \sum_{i \in V} tf_i$$

- ▶ *avdl*: Average document length over collection
- ▶ Length normalization component:

$$B = \left((1 - b) + b \frac{dl}{av_dl} \right), \quad 0 \leq b \leq 1$$

$b = 1$ full document length normalization

- ▶ $b = 0$ no document length normalization



Okapi BM25

- ▶ Factor in the frequency of each term versus doc length:

$$RSV^{BM25} = \sum_{i \in q} c_i^{BM25}(tf_i);$$

$$c_i^{BM25}(tf_i) = \log \frac{N}{df_i} \times \frac{(k_1 + 1)tf_i}{k_1((1 - b) + b \frac{dl}{av_dl}) + tf_i}$$

- ▶ $tf_{i,d}$ is term freq of i in d
- ▶ L_d is length of d and L_{ave} is ave. doc length
- ▶ k_1 and b are tuning parameters

Okapi BM25

$$RSV^{BM25} = \sum_{i \in q} \log \frac{N}{df_i} \cdot \frac{(k_1 + 1)tf_i}{k_1((1-b) + b \frac{dl}{avdl}) + tf_i}$$

- ▶ k_1 controls term frequency scaling
 - ▶ $k_1 = 0$ is binary model; $k_1 = \text{large}$ is raw term frequency
- ▶ b controls doc length normalization
 - ▶ $b = 0$ is no length normalization; $b = 1$ is relative frequency (fully scale by doc length)
- ▶ Typically, k_1 is set around 1.2–2 and b around 0.75

Resources

- S. E. Robertson and K. Spärck Jones. 1976. Relevance Weighting of Search Terms. *Journal of the American Society for Information Sciences* 27(3): 129–146.
- C. J. van Rijsbergen. 1979. *Information Retrieval*. 2nd ed. London: Butterworths, chapter 6. [Most details of math]
<http://www.dcs.gla.ac.uk/Keith/Preface.html>
- N. Fuhr. 1992. Probabilistic Models in Information Retrieval. *The Computer Journal*, 35(3), 243–255. [Easiest read, with BNs]
- F. Crestani, M. Lalmas, C. J. van Rijsbergen, and I. Campbell. 1998. Is This Document Relevant? ... Probably: A Survey of Probabilistic Models in Information Retrieval. *ACM Computing Surveys* 30(4): 528–552.
<http://www.acm.org/pubs/citations/journals/surveys/1998-30-4/p528-crestani/>
- [Adds very little material that isn't in van Rijsbergen or Fuhr]