PURDUE UNIVERSITY | Department of Computer Science

CS47300: Web Information Search and Management

Prof. Chris Clifton
9 September 2020
Probabilistic Retrieval Models
Material adapted from course created by Dr. Luo Si, now leading Alibaba research group

PURDUE UNIVERSITY
Department of Computer Science

Probabilistic IR topics

- Classical probabilistic retrieval model
  - Probability ranking principle, etc.
  - Binary independence model (≈ Naïve Bayes)
  - (Okapi) BM25
- Bayesian networks for text retrieval
- Language model approach to IR
  - An important emphasis in recent work

- *Probabilistic methods have been a recurring theme in Information Retrieval*
  - *Traditionally: neat ideas, but didn't win on performance*

29

PURDUE UNIVERSITY
Department of Computer Science

The document ranking problem

- We have a collection of documents
- User issues a query
- A list of documents needs to be returned
- **Ranking method is the core of an IR system:**
  - **In what order do we present documents to the user?**
  - We want the "best" document to be first, second best second, etc.…
- **Idea: Rank by probability of relevance of the document w.r.t. information need**
  - P(R=1|document$_i$, query)

30

PURDUE UNIVERSITY
Department of Computer Science

Recall a few probability basics

- For events A and B:
- Bayes' Rule

$$p(A,B) = p(A \cap B) = p(A \mid B)p(B) = p(B \mid A)p(A)$$

$$p(A \mid B) = \frac{p(B \mid A)p(A)}{p(B)} = \frac{p(B \mid A)p(A)}{\sum_{X=A,\bar{A}} p(B \mid X)p(X)}$$

Prior

Posterior

- Odds:

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

31

PURDUE UNIVERSITY
Department of Computer Science

The Probability Ranking Principle (PRP)

"If a reference retrieval system's response to each request is a ranking of the documents 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)

32

PURDUE UNIVERSITY
Department of Computer Science

Probability Ranking Principle (PRP)

Let x represent a document in the collection.
Let R represent **relevance** of a document w.r.t. given (fixed) query and let **R=1** represent relevant and **R=0** not relevant**.**

Need to find $p(R=1|x)$ - probability that a document x is **relevant.**

$$p(R=1|x)=\frac{p(x|R=1)p(R=1)}{p(x)}$$

$$p(R=0|x)=\frac{p(x|R=0)p(R=0)}{p(x)}$$

$$p(R=0|x)+p(R=1|x)=1$$

$p(R=1)$,$p(R=0)$ - prior probability of retrieving a relevant or non-relevant document

$p(x|R=1)$, $p(x|R=0)$ - probability that if a relevant (not relevant) document is retrieved, it is x.

36

PURDUE UNIVERSITY
Department of Computer Science

Probability Ranking Principle (PRP)

- Simple case: no selection costs or other utility concerns that would differentially weight errors
- PRP in action: Rank all documents by *p(R=1|x)*
- Theorem: Using the PRP is optimal, in that it minimizes the loss (Bayes risk) under 1/0 loss
  - Provable if all probabilities correct, etc. [e.g., Ripley 1996]
- How do we compute all those probabilities?
  - Do not know exact probabilities, have to use estimates

37

PURDUE UNIVERSITY
Department of Computer Science

Probabilistic Ranking

- **Basic concept:**

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

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

  - *Van Rijsbergen*

41

PURDUE UNIVERSITY
Department of Computer Science

Binary Independence Model

- Traditionally used in conjunction with PRP
- **"Binary" = Boolean**: documents are represented as binary incidence vectors of terms:
  - $\vec{x} = (x_1, \ldots, x_n)$
  - $x_i = 1$ iff term i is present in document x.
- **"Independence"**: terms occur in documents independently
- *Different documents can be modeled as the same vector*

PURDUE UNIVERSITY
Department of Computer Science

Binary Independence Model

- Queries: binary term incidence vectors
- Given query ***q***,
  - for each document ***d*** need to compute ***p(R|q,d).***
  - replace with computing ***p(R|q,x)*** where ***x*** is binary term incidence vector representing ***d.***
  - Interested only in ranking
- Use odds and Bayes' Rule:

$$O(R \mid q, \vec{x}) = \frac{p(R=1 \mid q, \vec{x})}{p(R=0 \mid q, \vec{x})} = \frac{\frac{p(R=1 \mid q)p(\vec{x} \mid R=1, q)}{p(\vec{x} \mid q)}}{\frac{p(R=0 \mid q)p(\vec{x} \mid R=0, q)}{p(\vec{x} \mid q)}}$$

PURDUE UNIVERSITY
Department of Computer Science

Binary Independence Model

$$O(R \mid q, \vec{x}) = \frac{p(R=1 \mid q, \vec{x})}{p(R=0 \mid q, \vec{x})} = \frac{p(R=1 \mid q)}{p(R=0 \mid q)} \times \frac{p(\vec{x} \mid R=1, q)}{p(\vec{x} \mid R=0, q)}$$

Constant for a given query

Needs estimation

- Using **Independence** Assumption:

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

$$O(R \mid q, \vec{x}) = O(R \mid q) \times \prod_{i=1}^{n} \frac{p(x_i \mid R=1, q)}{p(x_i \mid R=0, q)}$$

44

PURDUE UNIVERSITY
Department of Computer Science

Binary Independence Model

$$O(R \mid q, \vec{x}) = O(R \mid q) \times \prod_{i=1}^{n} \frac{p(x_i \mid R=1, q)}{p(x_i \mid R=0, q)}$$

- Since x_i is either 0 or 1:

$$O(R \mid q, \vec{x}) = O(R \mid q) \times \prod_{x_i=1} \frac{p(x_i=1 \mid R=1, q)}{p(x_i=1 \mid R=0, q)} \times \prod_{x_i=0} \frac{p(x_i=0 \mid R=1, q)}{p(x_i=0 \mid R=0, q)}$$

- Let $p_i = p(x_i=1 \mid R=1, q);\ r_i = p(x_i=1 \mid R=0, q);$

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

$$O(R \mid q, \vec{x}) = O(R \mid q) \times \prod_{\substack{x_i=1 \\ q_i=1}} \frac{p_i}{r_i} \times \prod_{\substack{x_i=0 \\ q_i=1}} \frac{(1-p_i)}{(1-r_i)}$$

45

PURDUE UNIVERSITY
Department of Computer Science

Model Parameters

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

PURDUE UNIVERSITY
Department of Computer Science

Binary Independence Model

$$O(R \mid q, \vec{x}) = O(R \mid q) \times \prod_{x_i=q_i=1} \frac{p_i}{r_i} \times \prod_{\substack{x_i=0 \\ q_i=1}} \frac{1-p_i}{1-r_i}$$

All matching terms

Non-matching query terms

$$O(R \mid q, \vec{x}) = O(R \mid q) \times \prod_{\substack{x_i=1 \\ q_i=1}} \frac{p_i}{r_i} \times \prod_{\substack{x_i=1 \\ q_i=1}} \frac{1-r_i}{1-p_i} \times \left(\prod_{\substack{x_i=0 \\ q_i=1}} \frac{1-p_i}{1-r_i} \right)$$

$$O(R \mid q, \vec{x}) = O(R \mid q) \times \prod_{x_i=q_i=1} \frac{p_i(1-r_i)}{r_i(1-p_i)} \times \prod_{q_i=1} \frac{1-p_i}{1-r_i}$$

All matching terms

All query terms

PURDUE UNIVERSITY
Department of Computer Science

Binary Independence Model

$$O(R \mid q, \vec{x}) = O(R \mid q) \cdot \prod_{x_i = q_i = 1} \frac{p_i(1-r_i)}{r_i(1-p_i)} \cdot \prod_{q_i = 1} \frac{1-p_i}{1-r_i}$$

Constant for each query

Only quantity to be estimated for rankings

Retrieval Status Value:

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

PURDUE UNIVERSITY
Department of Computer Science

Binary Independence Model

All boils down to computing RSV.

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

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

The c_i are log odds ratios
They function as the term weights in this model

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

PURDUE UNIVERSITY
Department of Computer Science

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	$n-s$	n
$x_i=0$	$S-s$	$N-n-S+s$	$N-n$
Total	S	$N-S$	N

For now, assume no zero terms.

- Estimates: $p_i \approx \frac{s}{S}$, $r_i \approx \frac{n-s}{N-S}$

$$c_i \approx K(N,n,S,s) = \log \frac{s/(S-s)}{(n-s)/(N-n-S+s)}$$

51

PURDUE UNIVERSITY
Department of Computer Science

Estimation – key challenge

- If non-relevant documents are approximated by the whole collection, then r_i (prob. of occurrence in non-relevant documents for query) *is n/N* and

$$\log \frac{1-r_i}{r_i} = \log \frac{N-n-S+s}{n-s} \approx \log \frac{N-n}{n} \approx \log \frac{N}{n} = IDF!$$

52

PURDUE UNIVERSITY
Department of Computer Science

Estimation – key challenge

- p_i (probability of occurrence in relevant documents) cannot be approximated as easily
- p_i can be estimated in various ways:
  - from relevant documents if know some
    - Relevance weighting can be used in a feedback loop
  - constant (Croft and Harper combination match) – then just get idf weighting of terms (with p_i=0.5)

$$RSV = \sum_{x_i=q_i=1} \log \frac{N}{n_i}$$

  - proportional to prob. of occurrence in collection
    - Greiff (SIGIR 1998) argues for 1/3 + 2/3 df_i/N