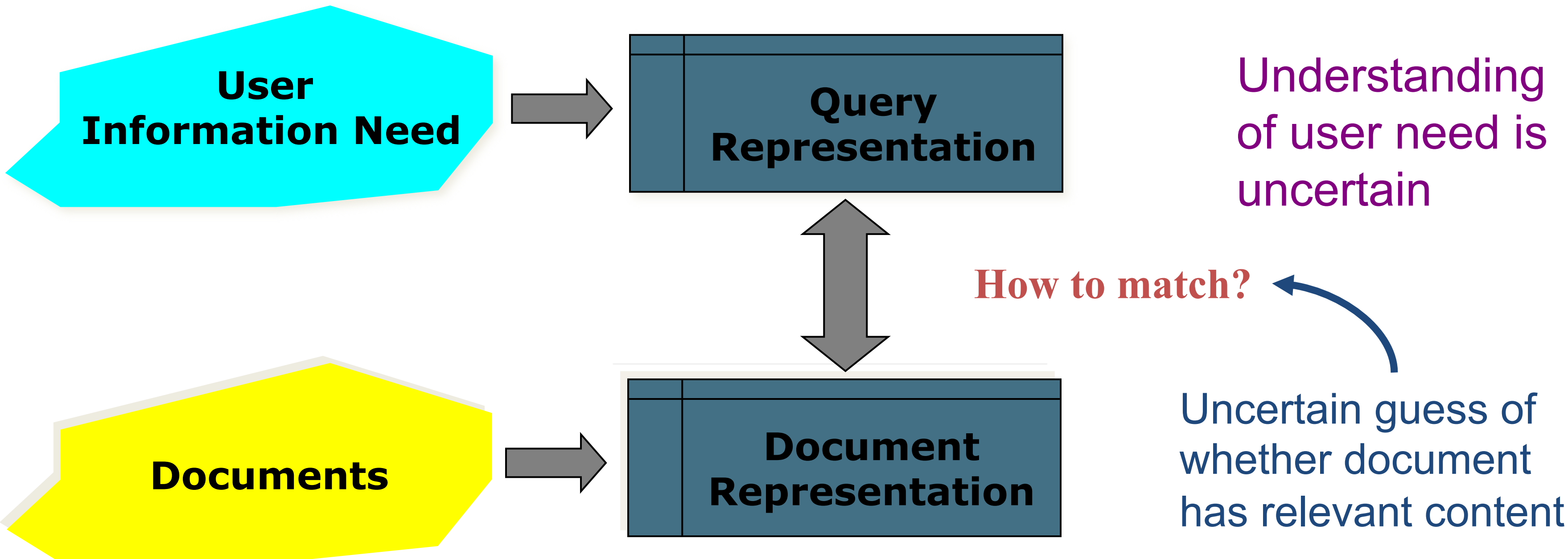
Introduction to

Information Retrieval

Lecture 11

Probabilistic Information Retrieval

Why probabilities in IR?



In vector space model (VSM), matching between each document 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 topics

- Classical probabilistic retrieval model
  - Probability ranking principle, etc.
  - Binary independence model (≈ Naïve Bayes text cat)
  - (Okapi) BM25

- Language model approach to IR
  - Next lecture

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 estimated probability of relevance of the document w.r.t. information need
  - $P(R=1|\text{document}_i, \text{query})$

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:

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

Probability Ranking Principle (PRP)

Let x represent a document in the collection.
Let R represent **relevance** of a document w.r.t. a 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;
Can rank documents in decreasing order of this probability.

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

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

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

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

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

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 for estimating the probabilities

Probability Ranking Principle (PRP)

- Questionable assumptions:
  - “Relevance” of each document is independent of relevance of other documents.
    - Really, it’s bad to keep on returning **duplicates**
  - “Term independence assumption”
    - Terms modeled as occurring in docs independently
    - Terms’ contributions to relevance treated as independent events
    - Similar to the ‘naïve’ assumption of the Naïve Bayes classifier
    - In a sense, equivalent to the assumption of a vector space model where each term is a dimension that is orthogonal to all other terms

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

Binary Independence Model

- Traditionally used in conjunction with PRP
- **“Binary” = Boolean**: documents are represented as binary incidence vectors of terms (cf. IIR Chapter 1):
  - $\vec{x} = (x_1, \ldots, x_n)$
  - $x_i = 1$ iff term *i* is present in document *x*, 0 otherwise.
- **“Independence”:** terms occur in documents independently (no association among terms is recognized)
- Different documents can be modeled as the same vector
- Queries are also represented as binary term incidence vectors

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:

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

Binary Independence Model

- 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
- Will use odds and Bayes' Rule:

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

Binary Independence Model

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

Prior probability of retrieving a relevant vs. non-relevant doc (constant for a given query)

This term only needs estimation

- Using **Independence** Assumption:

Presence / absence of a word is independent of the presence / absence of any other word in a document

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

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

Binary Independence Model

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

- Since x_i is either 0 or 1, we can separate the terms:

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

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

Probability of a term appearing in a relevant doc

Probability of a term appearing in a non-relevant doc

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

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

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

Binary Independence Model

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

Product over query terms found in the doc

over query terms not found in the doc

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

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

query terms found in the doc

All query terms

Summary Till Now

Odds of whether doc $\vec{x}$ is relevant w.r.t q

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

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

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

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

log odds ratio of a term i being relevant to q

Binary Independence Model

Prob any doc being relevant w.r.t q

log odds ratio of a term i being relevant to q

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

Odds of whether doc $\vec{x}$ is relevant w.r.t q

Constant for each query

Only quantity to be estimated for rankings

Retrieval Status Value:

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

Binary Independence Model

All boils down to computing RSV.

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

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

Ratio of
- Odds of a term appearing if a doc is relevant
- Odds of the term appearing if a doc is non-relevant

Positive if a term is more likely to appear in relevant documents

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

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

Binary Independence Model: probability estimates in theory

- 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

Number of docs that contain term x_i

- Estimates:

$$p_i \approx \frac{s}{S} \qquad r_i \approx \frac{(n-s)}{(N-S)}$$

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

For now, assume no zero terms. See book for **smoothing**.

Binary Independence Model: probability estimates in practice

- Assume that the relevant documents (or set of documents containing a particular term) are a very small fraction of the collection
- So 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

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

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

Relevance feedback in probabilistic IR

Probabilistic Relevance Feedback

1. Guess a preliminary probabilistic description of *R=1* documents (initial values for p_i and r_i) and use it to retrieve a first set of documents
2. Interact with the user to learn some definite members with R=1 and R=0 (relevance feedback for some subset of documents *V*)
3. Re-estimate p_i and r_i on the basis of the set *V*
   - Or can combine new information with original guess (use Bayesian prior):

$$p_i^{(2)} = \frac{|V_i| + \kappa p_i^{(1)}}{|V| + \kappa}$$

κ is prior weight

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

Iteratively estimating p_i *and* r_i (= Pseudo-relevance feedback)

1. Assume that p_i *is* constant over all x_i in query and r_i as before
   - p_i = 0.5 (even odds) for any given doc
2. Determine guess of relevant document set:
   - *V* is fixed size set of highest ranked documents on this model – get relevance feedback for the docs in *V*
3. We need to improve our guesses for p_i and r_i, so
   - Use distribution of x_i in docs in V. Let V_i be set of documents containing x_i
     - $p_i = |V_i| / |V|$
   - Assume if not retrieved then not relevant
     - $r_i = (n_i – |V_i|) / (N – |V|)$
4. Go to 2. until converges then return ranking

Okapi BM25: A non-binary model

Okapi BM25: A Non-binary Model

- The Binary Independence Model (BIM) was originally designed for short catalog records of fairly consistent length, and it works reasonably in these contexts

- For modern full-text search collections, a model should pay attention to term frequency and document length

- BestMatch25 (a.k.a BM25 or Okapi) is designed to be sensitive to these quantities.
- From 1994 until today, BM25 is one of the most widely used and robust retrieval models.

Okapi BM25: A Non-binary Model

- The simplest score for document d is just idf weighting of the query terms present in the document:

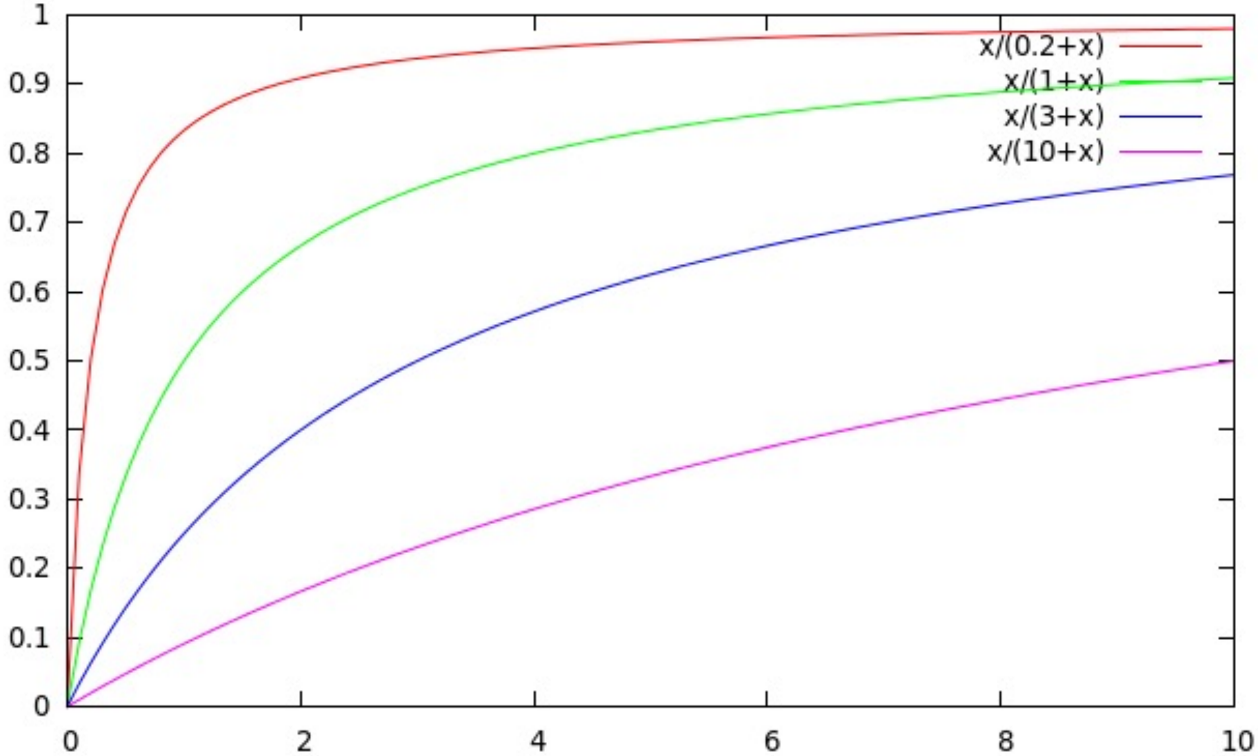
$$RSV_d = \sum_{t \in q} \log \frac{N}{\mathrm{df}_t}$$

- Improve this formula by factoring in the term frequency and document length:

$$RSV_d = \sum_{t \in q} \log \left[\frac{N}{\mathrm{df}_t}\right] \cdot \frac{(k_1 + 1)\mathrm{tf}_{td}}{k_1((1-b) + b \times (L_d/L_{ave})) + \mathrm{tf}_{td}}$$

- tf_{td} : term frequency in document d
- L_d (L_{ave}): length of document d (average document length in the whole collection)
- k_1: tuning parameter controlling the document term frequency scaling (0 means no TF, i.e., Binary model; large value means use raw TF)
- b: tuning parameter controlling the scaling by document length (0 means no length normalization)

Saturation function



- For high values of k_1, increments in tf_i continue to contribute significantly to the score
- Contributions tail off quickly for low values of k_1

Okapi BM25: A Non-binary Model

- If the query is long, we might also use similar weighting for query terms

$$RSV_d = \sum_{t \in q} \left[\log \frac{N}{\mathrm{df}_t} \right] \cdot \frac{(k_1 + 1)\mathrm{tf}_{td}}{k_1((1-b) + b \times (L_d/L_{\mathrm{ave}})) + \mathrm{tf}_{td}} \cdot \frac{(k_3 + 1)\mathrm{tf}_{tq}}{k_3 + \mathrm{tf}_{tq}}$$

- tf $_{tq}$: term frequency in the query q
- k_3: tuning parameter controlling term frequency scaling of the query
- No length normalisation of queries (because retrieval is being done with respect to a single fixed query)
- The above tuning parameters should ideally be set to optimize performance on a development test collection. In the absence of such optimisation, experiments have shown reasonable values are to set k_1 and k_3 to a value between 1.2 and 2 and b = 0.75

Probabilistic IR topics

- Classical probabilistic retrieval model
  - Probability ranking principle, etc.
  - Binary independence model (≈ Naïve Bayes text cat)
  - (Okapi) BM25

- Language model approach to IR
  - Next lecture