SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE NAME : 19CS732 INFORMATION RETRIEVAL TECHNIQUES

IVYEAR / VII SEMESTER

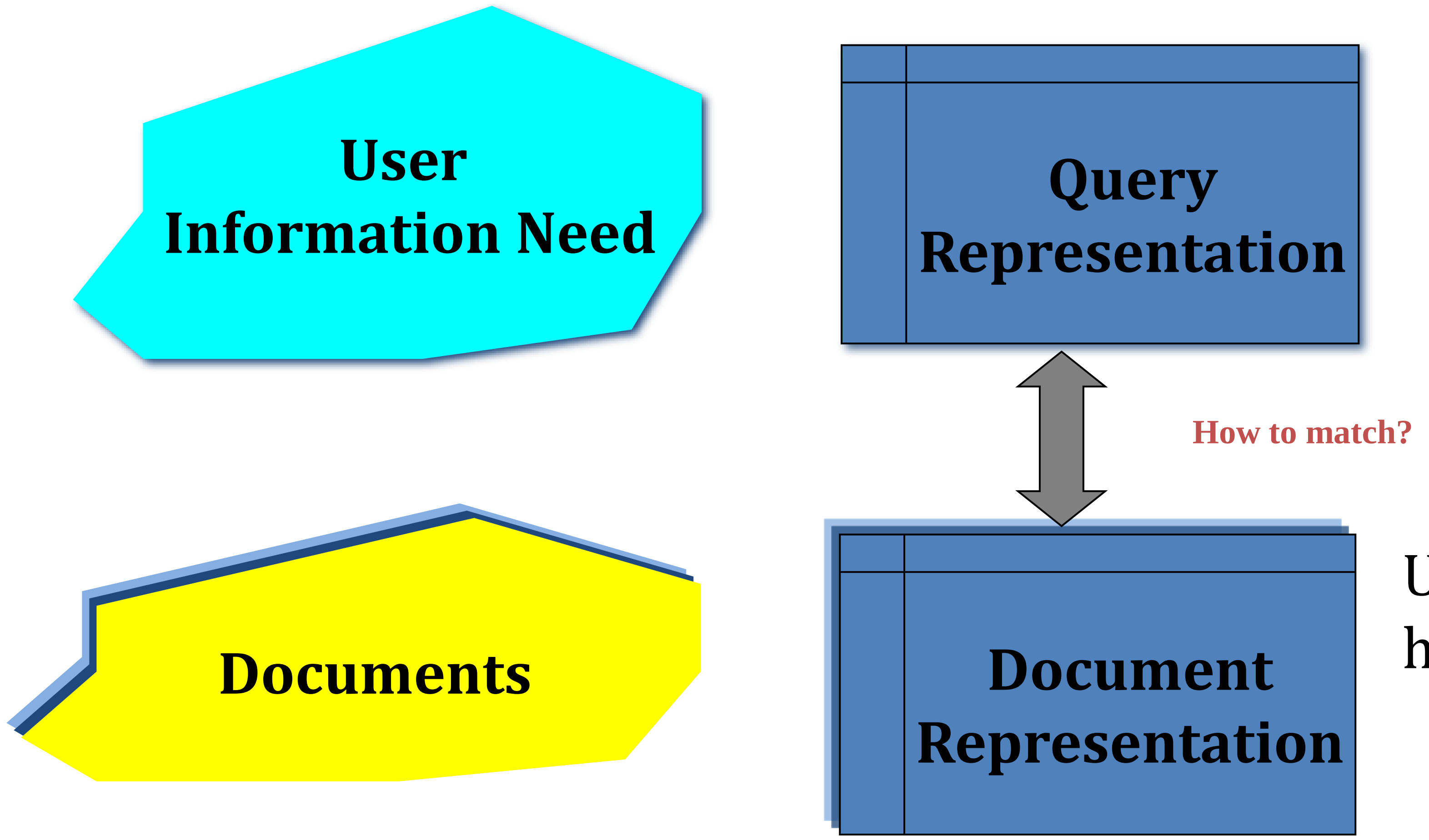
Unit 2- MODELING AND RETRIEVAL EVALUATION

Topic 4 : Probabilistic Model

Problem

- How to determine important words in a document?
  - Word sense?
  - Word *n*-grams (and phrases, idioms,…) → terms
- How to determine the degree of importance of a term within a document and within the entire collection?
- How to determine the degree of similarity between a document and the query?
- In the case of the web, what is the collection and what are the effects of links, formatting information, etc.?

Why probabilities in IR?



Understanding of user need is uncertain

Uncertain guess of whether document has relevant content

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?

Probabilities in IR

- Classical probabilistic retrieval model
  - Probability ranking principle, etc.
  - Binary independence model (≈ Naïve Bayes text cat)
  - (Okapi) BM25
- Bayesian networks for text retrieval
- Language model approach to IR
  - An important emphasis in recent work
- *Probabilistic methods are one of the oldest but also one of the currently hottest topics in IR.*

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

- $P(R=1|\text{document}_i, \text{query})$

Recall a few probability basics

- For events *A* and *B:*
- Bayes' Rule

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

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

- Odds:

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

Probability Ranking Principle

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

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

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

Probability Ranking Principle

- 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
- 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' contributions to relevance are treated as independent events.

Probabilistic Retrieval Strategy

- Estimate how terms contribute to relevance
  - How do things like tf, df, and document length influence your judgments about document relevance?
    - A more nuanced answer is the Okapi formulae
      - Spärck Jones / Robertson
- Combine to find document relevance probability
- Order documents by decreasing probability

Probability Ranking Principle-Cont..

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*.
- **"Independence":** terms occur in documents independently
- Different documents can be modeled as the same vector

Probability Ranking Principle-Cont..

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
- Will use odds and Bayes' Rule:

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

Probability Ranking Principle-Cont..

Binary Independence Model

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

Constant for a given query

Needs estimation

- Using **Independence** Assumption:

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

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

Probability Ranking Principle-Cont..

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:

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

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

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

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

Binary Independence Model

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

All matching terms

Non-matching query terms

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

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

All matching terms

All query terms

Binary Independence Model

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

Constant for each query

Only quantity to be estimated for rankings

Retrieval Status Value:

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

Probability Ranking Principle-Cont..

Binary Independence Model

All boils down to computing RSV.

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

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

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 ?

Probability Ranking Principle-Cont..

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

- Estimates: $p_i \approx \frac{s}{S}$ $\quad r_i \approx \frac{(n-s)}{(N-S)}$

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

For now, assume no zero terms. See later lecture.

Probability Ranking Principle-Cont..

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

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

Probability Ranking Principle-Cont..

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

Activity

Disadvantages

- Need to guess the initial ranking
- Binary weights, ignores frequencies
- Independence assumption (not clear if bad)

Advantages

- Theoretical adequacy: ranks by probabilities

Assessment 1

1. List out the Advantages of Probability in IRT
   a)________________________________
   b)________________________________
   c)________________________________
   d)________________________________

2. Identify the disadvantages of Probability in IRT
   a)________________________________
   b)________________________________
   c)________________________________
   d)________________________________

TEXT BOOKS:

1. Ricardo Baeza-Yates and Berthier Ribeiro-Neto, ―Modern Information Retrieval: The Concepts and Technology behind Search, Second Edition, ACM Press Books, 2011.

2. Ricci, F, Rokach, L. Shapira, B.Kantor, ―Recommender Systems Handbook‖, First Edition, 2011.

REFERENCES:

1. C. Manning, P. Raghavan, and H. Schütze, ―Introduction to Information Retrieval, Cambridge University Press, 2008.

2. Stefan Buettcher, Charles L. A. Clarke and Gordon V. Cormack, ―Information Retrieval: Implementing and Evaluating Search Engines, The MIT Press, 2010.

THANK YOU