

Lecture 4

Probabilistic Information Retrieval

Contents

- Basic Probability Conception
- Basic Probabilistic Principle
- Binary Independence Retrieval

Probability: independent random variables and conditional probability

Notation

Let a , b be two events, with probability $P(a)$ and $P(b)$.

Independent events

The events a and b are independent if and only if:

$$P(a \cap b) = P(b) P(a)$$

Conditional probability

$P(a \mid b)$ is the probability of a given b , also called the conditional probability of a given b .

$$P(a \mid b) P(b) = P(a \cap b) = P(b \mid a) P(a)$$

Probability Theory -- Bayesian Formulas

Notation

Let a , b be two events.

$P(a \mid b)$ is the probability of a given b

Bayes Theorem

$$P(a \mid b) = \frac{P(b \mid a) P(a)}{P(b)}$$

$$P(\bar{a} \mid b) = \frac{P(b \mid \bar{a}) P(\bar{a})}{P(b)} \quad \text{where } \bar{a} \text{ is the event } \textit{not } a$$

Contents

- *Basic Probability Conception*
- Basic Probabilistic Principle
- Binary Independence Retrieval

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

Concept

R is a set of documents that are guessed to be relevant

NR is the complement of R .

1. Guess a preliminary probabilistic description of R and use it to retrieve a first set of documents.
2. Interact with the user to refine the description.
3. Repeat, thus generating a succession of approximations to R .

Probabilistic Principle

Basic concept:

The probability that a document is relevant to a query is assumed to depend on the **terms in the query** and the **terms used to index the document**, only.

Given a user query q , the **ideal answer set**, R , is the set of all relevant documents.

Given a user query q and a document d_j in the collection, the probabilistic model **estimates the probability** that the user will find d_j relevant, i.e., that d_j is a member of R .

Tasks of Ranking by Probability

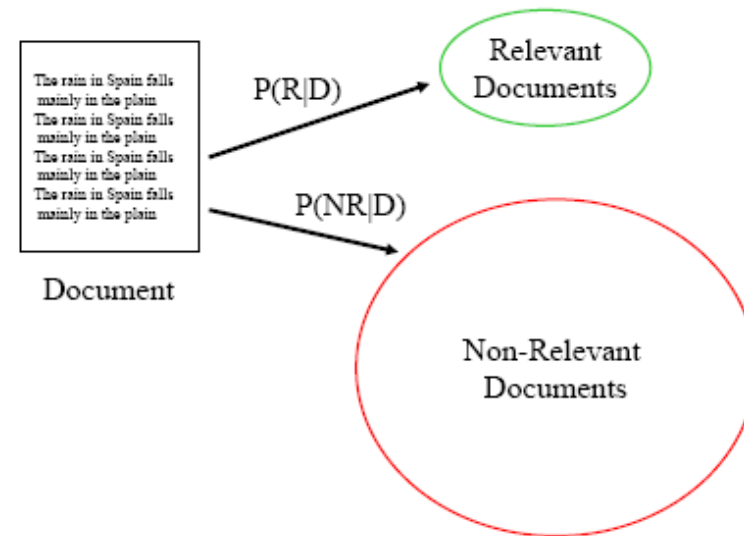
- 1. Find out relevant documents.
- 2. Construct the probability estimation model.
- 3. Design the query-document similarity measure based on probability of a document belonging to the relevant set.
- Our introduction will take at the inverse order, i.e., similarity measure->probability model -> relevant documents

Probabilistic Principle

Similarity measure:

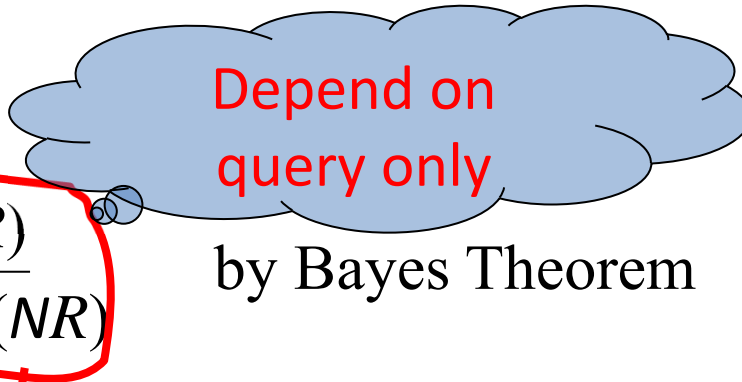
The *similarity* (d_j, q) is the ratio of the probability that d_j is relevant to q , to the probability that d_j is not relevant to q .

This measure runs from near zero, if the probability is small that the document is relevant, to large as the probability of relevance approaches one.



Probabilistic Principle

Given a query q and a document d_j the model needs an estimate of the **probability** that the user finds d_j relevant. i.e., $P(R | d_j)$.

$$\begin{aligned} \text{similarity}(d_j, q) &= \frac{P(R | d_j)}{P(NR | d_j)} \\ &= \frac{P(d_j | R) P(R)}{P(d_j | NR) P(NR)} \\ &= \frac{P(d_j | R)}{P(d_j | NR)} \times k \end{aligned}$$


Depend on query only

by Bayes Theorem

where k is constant

$P(d_j | R)$ is the probability of randomly selecting d_j from R .

Basic Probabilistic Principle-Risk Minimization

- Bayes Decision Rule: Retrieve if $P(R | D) > P(NR | D)$
 - minimizes the average probability of error:
$$P(\text{error} | D) = \begin{cases} P(R | D) & \text{if we decide NR} \\ P(NR | D) & \text{if we decide R} \end{cases}$$
 - equivalent to optimizing miss/ fallout tradeoff
- In general we can associate a *cost* with each type of error
- Derive a decision rule that minimizes *risk*
- Expected loss when deciding on class is *conditional risk*

Risk Minimization

- Risk of deciding *relevant* given D is
 - $c_{rr}P(R|D) + c_{rn}P(NR|D)$
 - c_{rr} is cost of deciding relevant when is relevant
 - c_{rn} is cost of deciding relevant when not relevant
- similarly, risk of deciding *non-relevant* is
 - $c_{nr}P(R|D) + c_{nn}P(NR|D)$
- How to decide the values of c_{rr} , c_{rn} , c_{nr} and c_{nn} ?

Risk Minimization (Cont'd)

- Minimizing risk gives
 - Want risk of choosing relevant to be less than that of choosing non-relevant
 - i.e., want that $c_{rr}P(R|D) + c_{rn}P(NR|D) < c_{nr}P(R|D) + c_{nn}P(NR|D)$
 - $(c_{nr}-c_{rr})P(R|D) > (c_{rn}-c_{nn})P(NR|D)$
 - So want $O(R|D) = P(R|D) / P(NR|D) > (c_{rn}-c_{nn})/(c_{nr}-c_{rr})$
- As always, hard to estimate, so transform with Bayes' theorem, so...
 - $(c_{nr}-c_{rr}) P(D|R)P(R) > (c_{rn}-c_{nn}) P(D|NR)P(NR)$ (P(D) cancels)
- Likelihood ratio
 - $P(D|R)/P(D|NR) > (c_{rn}-c_{nn})P(NR)/(c_{nr}-c_{rr}) P(R)$
 - retrieve when likelihood ratio greater than some threshold or, rank by likelihood ratio
 - typically $c_{rr} = c_{nn} = 0$ and $c_{rn} = c_{nr} = 1$
- Tasks left are the estimations of $P(D|R)$ and $P(D|NR)$.

Contents

- *Basic Probability Conception*
- *Basic Probabilistic Principle*
- **Binary Independence Retrieval**

Binary Independence Retrieval (BIR)

- Assume document D is represented by a binary vector $\mathbf{d} = (d_1, d_2, \dots, d_n)$ where $d_i = 0$ or 1 indicates the absence or presence of the index term
- $p_i = P(d_i = 1|R)$ $1 - p_i = P(d_i = 0|R)$
- $q_i = P(d_i = 1|NR)$ $1 - q_i = P(d_i = 0|NR)$
- Assume conditional independence, write $P(\mathbf{d}|R)$ as the product of the probabilities for the components of \mathbf{d} (i.e., product of probabilities of getting a particular vector of 1's and 0's)
- Likelihood functions are

$$P(\mathbf{d} | R) = \prod_{i=1}^n p_i^{d_i} (1 - p_i)^{1-d_i} \quad P(\mathbf{d} | NR) = \prod_{i=1}^n q_i^{d_i} (1 - q_i)^{1-d_i}$$

$$\frac{P(\mathbf{d}|R)}{P(\mathbf{d}|NR)} = \prod_{i=1}^n \left(\frac{p_i}{q_i} \right)^{d_i} \left(\frac{1-p_i}{1-q_i} \right)^{1-d_i} \quad (\text{sometimes called "linked dependence" assumption})$$

BIR (Cont'd)

- Convert to a linear discriminant function

If ranking derived from:
 $P(D|R)/P(D|NR) > P(NR)/P(R)$

- $g(\mathbf{d}) = \log P(\mathbf{d}|R)/P(\mathbf{d}|NR) + \log P(R)/P(NR)$

$$g(\mathbf{d}) = \sum_{i=1}^n \left[d_i \log \frac{p_i}{q_i} + (1 - d_i) \log \frac{1 - p_i}{1 - q_i} \right] + \log \frac{P(R)}{P(NR)}$$

$$g(\mathbf{d}) = \sum_{i=1}^n d_i \log \frac{p_i(1 - q_i)}{q_i(1 - p_i)} + \left[\sum_{i=1}^n \log \frac{1 - p_i}{1 - q_i} + \log \frac{P(R)}{P(NR)} \right]$$

$$P(\mathbf{d} | R) = \prod_{i=1}^n p_i^{d_i} (1 - p_i)^{1-d_i}$$

$$P(\mathbf{d} | NR) = \prod_{i=1}^n q_i^{d_i} (1 - q_i)^{1-d_i}$$

- The second term is a constant (d_i does not occur) for a given query and does not affect the ranking
- Assume that for terms not in the query, $p_i = q_i$
 - this can be changed, but is a handy simplification
- The *retrieval status value* for a document is computed by summing the weight $\log(p_i/(1-p_i)) + \log((1-q_i)/q_i)$ for all query terms in the document

Estimation using training data

- Must estimate p_i and q_i
- Given information about relevant documents, we would have the following contingency table:

	Relevant	Not Relevant	
$d_i = 1$	r	$n - r$	n
$d_i = 0$	$R - r$	$N - n - R + r$	$N - n$
	R	$N - R$	

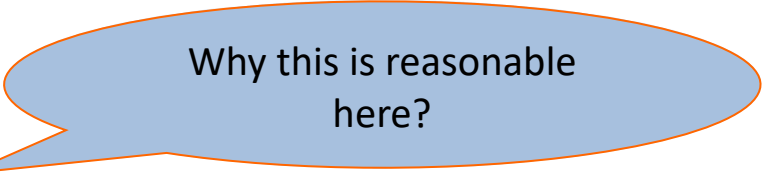
- Use maximum likelihood estimators
 - $p_i = P(d_i=1|R) = r/R$ $q_i = P(d_i=1|NR) = (n-r)/(N-R)$
 - Robertson and Sparck Jones F4 weight: $\log(r/(R-r))/((n-r)/(N-n-R+r))$
- Relevance information is usually not available
 - (except after/if searcher provides relevance feedback)
- Estimate probabilities based on information in the query, the document collection
 - previous queries can also be used with some learning approaches

Estimation without training data

Initial guess, with no information to work from:

$$p_i = P(d_i \mid R) = c$$

$$q_i = P(d_i \mid NR) = n_i / N$$



Why this is reasonable
here?

where:

c is an arbitrary constant, e.g., 0.5

n_i is the number of documents that contain term x_i

N is the total number of documents in the collection

Improving the Estimation

Human feedback -- relevance feedback

Automatically

(a) Run query q using initial values. Consider the t top ranked documents. Let s_i be the number of these documents that contain the term d_i .

(b) The new estimates are:

$$p_i = P(d_i \mid R) = s_i / t$$

$$q_i = P(d_i \mid NR) = (n_i - s_i) / (N - t)$$

Estimation issues

- Maximum likelihood estimates have problems with small samples or zero values (recall “smoothing”)
- Standard statistical practice is Bayesian estimates of the form

$$\hat{p} = \frac{x + a}{n + a + b}$$

- where x is the number of successes in n trials
 - a and b are parameters determined by the combination of assumptions about prior distributions and loss functions
- Typically use $a = b = 0.5$
- Estimating probabilities is the same problem as determining weighting formulae in less formal models

Adding more document representation

- BIR model assumes *binary* indexing of documents
 - Term is present or not
 - Experience shows that *tf* conveys information
 - IDF appears in some form in BIR as shown
- Want to incorporate term weights into model
- Could just multiply by the term weight (e.g., *tf*)

$$\boxed{g(d) = \sum_{i=1}^n d_i \log \frac{p_i(1 - q_i)}{q_i(1 - p_i)}} \quad \longrightarrow \quad \boxed{g'(d) = \sum_{i=1}^n w_{d_i} d_i \log \frac{p_i(1 - q_i)}{q_i(1 - p_i)}}$$

- Could use information to adjust estimate of p_i
 - Gives greater range of possible values for p_i
- But neither of those is theoretically compelling
 - Extend model instead...

BM25 Weighting

BM25 Weighting

Ref for following pages:

Stephen Robertson, Microsoft Research Cambridge, UK, CCF ADL 2011

document length $dl := \sum_V tf_i$
average doclength $avdl$ average over collection

- Soft normalization factor:

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

Now divide tf by B before applying the saturation function

$$tf'_i = \frac{tf_i}{B}$$

$$\begin{aligned} W_i^{\text{BM25}} &= \frac{tf'_i}{k_1 + tf'_i} W_i^{\text{RSJ}} \\ &= \frac{tf_i}{k_1 \left((1 - b) + b \frac{dl}{avdl} \right) + tf_i} W_i^{\text{RSJ}} \end{aligned}$$

W_i^{RSJ} denotes the Robertson / Sparck Jones F4 weight

Here k_1 is just another constant.

Multiple fields

- Assume multiple *fields* (also known as *streams*) in documents
 - E.g. title, abstract, body
 - Basically a minimal flat structure, common to all documents
- Assume also that some fields may provide better evidence about relevance than others
 - Would like to weight them differentially

fields	s	$= 1, \dots, S$
field lengths	sl_s	
field weights	w_s	
document	$(\mathbf{tf}_1 \dots \mathbf{tf}_V)$	vectors
\mathbf{tf}_i vector	$(tf_{1i}, \dots, tf_{Si})$	

Multiple fields

- One possibility: Calculate BM25 per field and then combine
- But this is **not** a good idea
 - each field calculation ignores the other fields
 - and in particular which terms were matched
- Alternative: look at each **term** in turn
 - combine information across fields for this term
 - then apply saturation function
 - and combine across terms
- Field weights as field replication
 - E.g. if terms in the title field should count 5 times as much as terms in the body
 - ... then simply replicate the title field 5 times

BM25F

- Simplest version:
$$\widetilde{tf}_i = \sum_{s=1}^S w_s tf_{si}$$
$$\widetilde{dl} = \sum_{s=1}^S w_s sl_s$$
- But we may want to use field-specific BM25 parameters
 - Could consider k_1 , b , W_i^{RSJ} , or its non-feedback version, W_i^{IDF}
 - BM25 formula can be re-arranged to combine any or all of these with the field-specific tf
 - Have found it important to make b field-specific

BM25F

$$\widetilde{tf}_i = \sum_{s=1}^S w_s \frac{tf_{si}}{B_s}$$

$$B_s = \left((1 - b_s) + b_s \frac{sl_s}{avsl_s} \right), \quad 0 \leq b_s \leq 1$$

$$W_i^{\text{BM25F}} = \frac{\widetilde{tf}_i}{k_1 + \widetilde{tf}_i} W_i^{\text{RSJ}}$$

This is the version used in TREC submissions (the CIKM paper [Robertson Zaragoza Taylor 2004] used the simpler version)

Again, in the absence of relevance feedback information, w_i^{RSJ} can be replaced by w_i^{IDF}

Homework 04