Information Retrieval

Weike Pan

The slides are **adapted from those provided by Prof. Hinrich Schütze** at University of Munich (http://www.cis.lmu.de/~hs/teach/14s/ir/).

Background

- Given a user's information need (represented as a query) and a collection of documents (transformed into document representations), a system must determine how well the documents satisfy the query

- An IR system has an **uncertain** understanding of the user query, and makes an **uncertain** guess of whether a document satisfies the query

- Probability theory provides a principled foundation for such reasoning under **uncertainty**
- Probabilistic models exploit this foundation to estimate how likely it is that a document is relevant to a query

Chapter 11 Probabilistic information retrieval

- 11.1 Review of basic probability theory
- 11.2 The Probability Ranking Principle
- 11.3 The Binary Independence Model
- 11.4 An appraisal and some extensions
- 11.5 References and further reading

Outline

- 11.1 Review of basic probability theory
- 11.2 The Probability Ranking Principle
- 11.3 The Binary Independence Model
- 11.4 An appraisal and some extensions
- 11.5 References and further reading

11.1 Review of basic probability theory

- For two events A and B, the joint event of both events occurring is described by the joint probability $P(A,B)$.

- The conditional probability $P(A|B)$ expresses the probability of event A given that event B occurred.

- Chain rule:

$$P(A,B) = P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$$

- Partition rule:

$$P(B) = P(A,B) + P(\overline{A},B)$$

11.1 Review of basic probability theory

- Bayes rule

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \left[\frac{P(B|A)}{\sum_{X\in\{A,\overline{A}\}} P(B|X)P(X)}\right] P(A)$$

- Odds of an event A

$$O(A) = \frac{P(A)}{P(\overline{A})} = \frac{P(A)}{1 - P(A)}$$

Outline

- 11.1 Review of basic probability theory
- 11.2 The Probability Ranking Principle
- 11.3 The Binary Independence Model
- 11.4 An appraisal and some extensions
- 11.5 References and further reading

11.2 The Probability Ranking Principle

- Ranked retrieval setup: given a collection of documents, the user issues a query, and an ordered list of documents is returned

- Assume binary notion of relevance:
  - $R_{d,q} = 1$ if document d is relevant w.r.t. query q
  - $R_{d,q} = 0$ otherwise

- Rank documents decreasingly by their estimated probability of relevance w.r.t. the information need: $P(R = 1|d, q)$
  - This is the basis of the **Probability Ranking Principle (PRP)**

Outline

- 11.1 Review of basic probability theory
- 11.2 The Probability Ranking Principle
- 11.3 The Binary Independence Model
- 11.4 An appraisal and some extensions
- 11.5 References and further reading

11.3 The Binary Independence Model

- Assumptions:
  - **'Binary'** (equivalent to Boolean): documents and queries represented as binary term incidence vectors
    - E.g., document d represented by vector $\vec{x} = (x_1, \ldots, x_M)$, where $x_t = 1$ if term t occurs in d and $x_t = 0$ otherwise
    - Different documents may have the same vector representation
  - **'Independence'**: no association between terms

11.3.1 Deriving a ranking function for query terms

- Given a query q, ranking documents by $P(R=1|d,q)$ is modeled under **BIM** as ranking them by $P(R=1|\vec{x},\vec{q})$
- Rank documents by their odds of relevance (gives the same ranking)

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

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

(constant given a query)

$P(\vec{x}|R=1,\vec{q})$ Probability that if a relevant document is retrieved, then that document's representation is $\vec{x}$

11.3.1 Deriving a ranking function for query terms

- Based on the independence assumption and the binary assumption, we have,

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

independence assumption

$$= \prod_{t=1}^{M} \frac{P(x_t|R=1,\vec{q})}{P(x_t|R=0,\vec{q})}$$

binary assumption

$$= \prod_{t:x_t=1} \frac{P(x_t=1|R=1,\vec{q})}{P(x_t=1|R=0,\vec{q})} \cdot \prod_{t:x_t=0} \frac{P(x_t=0|R=1,\vec{q})}{P(x_t=0|R=0,\vec{q})}$$

11.3.1 Deriving a ranking function for query terms

- Based on the assumption that terms not occurring in the query are equally likely to occur in relevant and nonrelevant documents, we have,

$$O(R|\vec{x},\vec{q}) \propto \prod_{t:x_t=1,q_t=1} \frac{P(x_t=1|R=1,\vec{q})}{P(x_t=1|R=0,\vec{q})} \cdot \prod_{t:x_t=0,q_t=1} \frac{P(x_t=0|R=1,\vec{q})}{P(x_t=0|R=0,\vec{q})}$$

简写

$$= \prod_{t:x_t=1,q_t=1} \frac{p_t}{u_t} \cdot \prod_{t:x_t=0,q_t=1} \frac{1-p_t}{1-u_t}$$

等价变换

$$= \prod_{t:x_t=1,q_t=1} \frac{p_t\boxed{(1-u_t)}}{u_t\boxed{(1-p_t)}} \cdot \underbrace{\prod_{t:q_t=1} \frac{1-p_t}{1-u_t}}_{\text{(constant for a query)}}$$

$$\propto \prod_{t:x_t=1,q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)}$$

where $p_t = P(x_t=1|R=1,\vec{q})$ and $u_t = P(x_t=1|R=0,\vec{q})$.

11.3.1 Deriving a ranking function for query terms

- The resulting quantity used for ranking is called the Retrieval Status Value (RSV) in this model

$$RSV_d = \log \prod_{t:x_t=q_t=1} \frac{p_t(1-u_t)}{u_t(1-p_t)} = \sum_{t:x_t=q_t=1} \log \frac{p_t(1-u_t)}{u_t(1-p_t)}$$

- We have

$$RSV_d = \textstyle\sum_{x_t=q_t=1} c_t$$

等价变换

$$c_t = \log \frac{p_t(1-u_t)}{u_t(1-p_t)} = \log \frac{p_t}{1-p_t} - \log \frac{u_t}{1-u_t}$$

- The odds of the term appearing if the document is relevant $(p_t/(1-p_t))$
- The odds of the term appearing if the document is nonrelevant $(u_t/(1-u_t))$

11.3.2 Probability estimates in theory

- For each term t, we have a contingency table (列联表) of counts of documents in the collection

	documents	relevant	nonrelevant	Total
Term present	$x_t = 1$	s	$\mathrm{df}_t - s$	df_t
Term absent	$x_t = 0$	$S - s$	$(N - \mathrm{df}_t) - (S - s)$	$N - \mathrm{df}_t$
	Total	S	$N - S$	N

Using this, $p_t = s/S$ and $u_t = (\mathrm{df}_t - s)/(N - S)$ and

$$c_t = K(N, \mathrm{df}_t, S, s) = \log \frac{s/(S-s)}{(\mathrm{df}_t - s)/((N - \mathrm{df}_t) - (S - s))}$$

代入可得

11.3.3 Probability estimates in practice

- Assuming that relevant documents are a very small percentage of the collection, approximate statistics for nonrelevant documents by statistics from the whole collection, we use df_t/N for u_t . We then have:

$$\log[(1-u_t)/u_t] = \log[(N-\mathrm{df}_t)/\mathrm{df}_t] \approx \log N/\mathrm{df}_t$$

- This becomes **IDF (inverse document frequency)**. In other words, we can provide **a theoretical justification** for the most frequently used form of IDF weighting.

11.3.3 Probability estimates in practice

- The quantity p_t can be estimated in various ways (不同的估计方法):
  - We can use the frequency of term occurrence in known relevant documents (if we know some). That is, we have some labeled data.
  - Croft and Harper (1979): $p_t = 0.5$. It is a constant.
  - Greiff (1998): $p_t = \frac{1}{3} + \frac{2}{3}\mathrm{df}_t / N$.
  - …

Exercise

- **Query**: Obama health plan
- **Doc1**: Obama rejects allegations about his own bad health Nonrelevant
- **Doc2**: The plan is to visit Obama Nonrelevant
- **Doc3**: Obama raises concerns with US health plan reforms Relevant

- Estimate the probability that the above documents are relevant to the query. These are the only three documents in the collection.

Outline

- 11.1 Review of basic probability theory
- 11.2 The Probability Ranking Principle
- 11.3 The Binary Independence Model
- 11.4 An appraisal and some extensions
- 11.5 References and further reading

11.4 An appraisal and some extensions

- Okapi BM25 basic weighting

$\boxed{\text{TF-IDF}}$

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

Here, tf_{td} is the frequency of term t in document d, and L_d and L_{ave} are the length of document d and the average document length for the whole collection. The variable k_1 is a positive tuning parameter that calibrates the document term frequency scaling. A k_1 value of 0 corresponds to a binary model (no term frequency), and a large value corresponds to using raw term frequency. b is another tuning parameter ($0 \leq b \leq 1$) which determines the scaling by document length: $b = 1$ corresponds to fully scaling the term weight by the document length, while $b = 0$ corresponds to no length normalization.

11.4 An appraisal and some extensions

- Okapi BM25 weighting for long queries

If the query is long, then we might also use similar weighting for query terms. This is appropriate if the queries are paragraph long information needs, but unnecessary for short queries.

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

with tf_{tq} being the frequency of term t in the query q, and k_3 being another positive tuning parameter that this time calibrates term frequency scaling of the query. In the equation presented, there is no length normalization of queries (it is as if $b = 0$ here). Length normalization of the query is unnecessary because retrieval is being done with respect to a single fixed query. The tuning parameters of these formulas should ideally be set to optimize performance on a development test collection (see page 153). That is, we can search for values of these parameters that maximize performance on a separate development test collection (either manually or with optimization methods such as grid search or something more advanced), and then use these parameters on the actual test collection. In the absence of such optimization, experiments have shown reasonable values are to set k_1 and k_3 to a value between 1.2 and 2 and $b = 0.75$.

11.4 An appraisal and some extensions

- Which ranking model should I use?
  - I want something basic and simple → use vector space with TF-IDF weighting.
  - I want to use a state-of-the-art ranking model with excellent performance → use language models or BM25 with tuned parameters

Summary

- 11.1 Review of basic probability theory
- 11.2 The Probability Ranking Principle
- 11.3 The Binary Independence Model
- 11.4 An appraisal and some extensions
- 11.5 References and further reading