Introduction to Information Retrieval

http://informationretrieval.org

IIR 11: Probabilistic Information Retrieval

Hinrich Schütze

Institute for Natural Language Processing, Universität Stuttgart

2011-08-29

Models and Methods

1. Boolean model and its limitations (30)
2. Vector space model (30)
3. Probabilistic models (30)
4. Language model-based retrieval (30)
5. Latent semantic indexing (30)
6. Learning to rank (30)

Take-away

Take-away

- Probabilistic approach to IR: Introduction

Take-away

- Probabilistic approach to IR: Introduction
- Binary independence model or BIM – the first influential probabilistic model

Take-away

- Probabilistic approach to IR: Introduction
- Binary independence model or BIM – the first influential probabilistic model
- Okapi BM25, a more modern, better performing probabilistic model

Outline

Probabilistic approach to IR

- The adhoc retrieval problem: Given a user information need and a collection of documents, the IR system must determine how well the documents satisfy the query.

Probabilistic approach to IR

- The adhoc retrieval problem: Given a user information need and a collection of documents, the IR system must determine how well the documents satisfy the query.
- The IR system has an uncertain understanding of the user query . . .

Probabilistic approach to IR

- The adhoc retrieval problem: Given a user information need and a collection of documents, the IR system must determine how well the documents satisfy the query.
- The IR system has an uncertain understanding of the user query ...
- ...and makes an uncertain guess of whether a document satisfies the query.

Probabilistic approach to IR

- The adhoc retrieval problem: Given a user information need and a collection of documents, the IR system must determine how well the documents satisfy the query.
- The 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 approach to IR

- The adhoc retrieval problem: Given a user information need and a collection of documents, the IR system must determine how well the documents satisfy the query.
- The 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 IR models exploit this foundation to estimate how likely it is that a document is relevant to a query.

Probabilistic vs. vector space model

Probabilistic vs. vector space model

- Vector space model: rank documents according to similarity to query.

Probabilistic vs. vector space model

- Vector space model: rank documents according to similarity to query.
- The notion of similarity does not translate directly into an assessment of "is the document a good document to give to the user or not?"

Probabilistic vs. vector space model

- Vector space model: rank documents according to similarity to query.
- The notion of similarity does not translate directly into an assessment of "is the document a good document to give to the user or not?"
- The most similar document can be highly relevant or completely nonrelevant.

Probabilistic vs. vector space model

- Vector space model: rank documents according to similarity to query.
- The notion of similarity does not translate directly into an assessment of "is the document a good document to give to the user or not?"
- The most similar document can be highly relevant or completely nonrelevant.
- Probability theory is arguably a cleaner formalization of what we really want an IR system to do: give relevant documents to the user.

Probabilistic IR models at a glance

Probabilistic IR models at a glance

- Classical probabilistic retrieval models

Probabilistic IR models at a glance

- Classical probabilistic retrieval models
  - Binary Independence Model

Probabilistic IR models at a glance

- Classical probabilistic retrieval models
  - Binary Independence Model
  - Okapi BM25

Probabilistic IR models at a glance

- Classical probabilistic retrieval models
  - Binary Independence Model
  - Okapi BM25
- Bayesian networks for text retrieval

Probabilistic IR models at a glance

- Classical probabilistic retrieval models
  - Binary Independence Model
  - Okapi BM25
- Bayesian networks for text retrieval
  - Don't have time for this

Probabilistic IR models at a glance

- Classical probabilistic retrieval models
  - Binary Independence Model
  - Okapi BM25
- Bayesian networks for text retrieval
  - Don't have time for this
- Language model approach to IR

Probabilistic IR models at a glance

- Classical probabilistic retrieval models
  - Binary Independence Model
  - Okapi BM25
- Bayesian networks for text retrieval
  - Don't have time for this
- Language model approach to IR
  - Important recent work, will be covered in the next lecture

Probabilistic IR and ranking

- Ranked retrieval setup: the user issues a query, and a ranked list of documents is returned.

Probabilistic IR and ranking

- Ranked retrieval setup: the user issues a query, and a ranked list of documents is returned.
- How can we rank probabilistically?

Probabilistic IR and ranking

- Ranked retrieval setup: the user issues a query, and a ranked list of documents is returned.
- How can we rank probabilistically?
- Let $R_{d,q}$ be a random dichotomous variable, such that

Probabilistic IR and ranking

- Ranked retrieval setup: the user issues a query, and a ranked list of documents is returned.
- How can we rank probabilistically?
- Let $R_{d,q}$ be a random dichotomous variable, such that
  - $R_{d,q} = 1$ if document d is relevant w.r.t query q

Probabilistic IR and ranking

- Ranked retrieval setup: the user issues a query, and a ranked list of documents is returned.
- How can we rank probabilistically?
- Let $R_{d,q}$ be a random dichotomous variable, such that
  - $R_{d,q} = 1$ if document d is relevant w.r.t query q
  - $R_{d,q} = 0$ otherwise

Probabilistic IR and ranking

- Ranked retrieval setup: the user issues a query, and a ranked list of documents is returned.
- How can we rank probabilistically?
- Let $R_{d,q}$ be a random dichotomous variable, such that
  - $R_{d,q} = 1$ if document d is relevant w.r.t query q
  - $R_{d,q} = 0$ otherwise
- (This is a binary notion of relevance.)

Probabilistic IR and ranking

- Ranked retrieval setup: the user issues a query, and a ranked list of documents is returned.
- How can we rank probabilistically?
- Let $R_{d,q}$ be a random dichotomous variable, such that
  - $R_{d,q} = 1$ if document d is relevant w.r.t query q
  - $R_{d,q} = 0$ otherwise
- (This is a binary notion of relevance.)
- Probabilistic ranking orders documents decreasingly by their estimated probability of relevance w.r.t. query: $P(R = 1|d, q)$

Probabilistic IR and ranking

- Ranked retrieval setup: the user issues a query, and a ranked list of documents is returned.
- How can we rank probabilistically?
- Let $R_{d,q}$ be a random dichotomous variable, such that
  - $R_{d,q} = 1$ if document d is relevant w.r.t query q
  - $R_{d,q} = 0$ otherwise
- (This is a binary notion of relevance.)
- Probabilistic ranking orders documents decreasingly by their estimated probability of relevance w.r.t. query: $P(R = 1|d, q)$
- How can we justify this way of proceeding? $\square$

Probability Ranking Principle (PRP)

If the retrieved documents are ranked decreasingly on their probability of relevance (w.r.t a query), then the effectiveness of the system will be the best that is obtainable.

Probability Ranking Principle (PRP)

If the retrieved documents are ranked decreasingly on their probability of relevance (w.r.t a query), then the effectiveness of the system will be the best that is obtainable.

Fundamental assumption: the relevance of each document is independent of the relevance of other documents.

Outline

Binary Independence Model (BIM)

- Binary: documents and queries represented as binary term incidence vectors

Binary Independence Model (BIM)

- Binary: documents and queries represented as binary term incidence vectors
- Independence: terms are independent of each other (not true, but works in practice – naive assumption of Naive Bayes models)

Binary incidence matrix

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
ANTHONY	1	1	0	0	0	1	
BRUTUS	1	1	0	1	0	0	
CAESAR	1	1	0	1	1	1	
CALPURNIA	0	1	0	0	0	0	
CLEOPATRA	1	0	0	0	0	0	
MERCY	1	0	1	1	1	1	
WORSER	1	0	1	1	1	0	
...							

Each document is represented as a binary vector $\in \{0, 1\}^{|V|}$.

Bayes' rule

Bayes' rule

- $$\begin{aligned} P(R=1|\vec{x},\vec{q}) &= \frac{P(\vec{x}|R=1,\vec{q})P(R=1|\vec{q})}{P(\vec{x}|\vec{q})} \\ P(R=0|\vec{x},\vec{q}) &= \frac{P(\vec{x}|R=0,\vec{q})P(R=0|\vec{q})}{P(\vec{x}|\vec{q})} \end{aligned}$$

Bayes' rule

-

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

- (Recall that document and query are modeled as term incidence vectors: $\vec{x}$ and $\vec{q}$.)

Bayes' rule

- $$\begin{aligned} P(R=1|\vec{x},\vec{q}) &= \frac{P(\vec{x}|R=1,\vec{q})P(R=1|\vec{q})}{P(\vec{x}|\vec{q})} \\ P(R=0|\vec{x},\vec{q}) &= \frac{P(\vec{x}|R=0,\vec{q})P(R=0|\vec{q})}{P(\vec{x}|\vec{q})} \end{aligned}$$
- (Recall that document and query are modeled as term incidence vectors: $\vec{x}$ and $\vec{q}$.)
- $P(\vec{x}|R=1,\vec{q})$ and $P(\vec{x}|R=0,\vec{q})$: probability that if a relevant or nonrelevant document is retrieved, then that document's representation is $\vec{x}$

Bayes' rule

- $$\begin{aligned} P(R=1|\vec{x},\vec{q}) &= \frac{P(\vec{x}|R=1,\vec{q})P(R=1|\vec{q})}{P(\vec{x}|\vec{q})} \\ P(R=0|\vec{x},\vec{q}) &= \frac{P(\vec{x}|R=0,\vec{q})P(R=0|\vec{q})}{P(\vec{x}|\vec{q})} \end{aligned}$$
- (Recall that document and query are modeled as term incidence vectors: $\vec{x}$ and $\vec{q}$.)
- $P(\vec{x}|R=1,\vec{q})$ and $P(\vec{x}|R=0,\vec{q})$: probability that if a relevant or nonrelevant document is retrieved, then that document's representation is $\vec{x}$
- Use statistics about the document collection to estimate these probabilities

Priors

$P(R|d,q)$ is modeled using term incidence vectors as $P(R|\vec{x},\vec{q})$

$$
\begin{aligned}
P(R=1|\vec{x},\vec{q}) &= \frac{P(\vec{x}|R=1,\vec{q})P(R=1|\vec{q})}{P(\vec{x}|\vec{q})} \\
P(R=0|\vec{x},\vec{q}) &= \frac{P(\vec{x}|R=0,\vec{q})P(R=0|\vec{q})}{P(\vec{x}|\vec{q})}
\end{aligned}
$$

Priors

$P(R|d,q)$ is modeled using term incidence vectors as $P(R|\vec{x},\vec{q})$

$$
\begin{aligned}
P(R=1|\vec{x},\vec{q}) &= \frac{P(\vec{x}|R=1,\vec{q})P(R=1|\vec{q})}{P(\vec{x}|\vec{q})} \\
P(R=0|\vec{x},\vec{q}) &= \frac{P(\vec{x}|R=0,\vec{q})P(R=0|\vec{q})}{P(\vec{x}|\vec{q})}
\end{aligned}
$$

- $P(R=1|\vec{q})$ and $P(R=0|\vec{q})$: prior probability of retrieving a relevant or nonrelevant document for a query $\vec{q}$

Priors

$P(R|d,q)$ is modeled using term incidence vectors as $P(R|\vec{x},\vec{q})$

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

- $P(R=1|\vec{q})$ and $P(R=0|\vec{q})$: prior probability of retrieving a relevant or nonrelevant document for a query $\vec{q}$
- Estimate $P(R=1|\vec{q})$ and $P(R=0|\vec{q})$ from percentage of relevant documents in the collection

Ranking according to odds

- We said that we're going to rank documents according to $P(R = 1|\vec{x}, \vec{q})$

Ranking according to odds

- We said that we're going to rank documents according to $P(R = 1|\vec{x}, \vec{q})$
- Easier: rank documents by their odds of relevance (gives same ranking)

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

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

Ranking according to odds

- We said that we're going to rank documents according to $P(R=1|\vec{x},\vec{q})$
- Easier: rank documents by their odds of relevance (gives same ranking)

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

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

- $\frac{P(R=1|\vec{q})}{P(R=0|\vec{q})}$ is a constant for a given query - can be ignored $\square$

Naive Bayes conditional independence assumption

Naive Bayes conditional independence assumption

Now we make the Naive Bayes conditional independence assumption that the presence or absence of a word in a document is independent of the presence or absence of any other word (given the query):

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

So:

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

Separating terms in the document vs. not

Since each x_t is either 0 or 1, we can separate the terms:

Separating terms in the document vs. not

Since each x_t is either 0 or 1, we can separate the terms:

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

Definition of p_t and u_t

- Let $p_t = P(x_t = 1 | R = 1, \vec{q})$ be the probability of a term appearing in relevant document.

Definition of p_t and u_t

- Let $p_t = P(x_t = 1|R = 1, \vec{q})$ be the probability of a term appearing in relevant document.
- Let $u_t = P(x_t = 1|R = 0, \vec{q})$ be the probability of a term appearing in a nonrelevant document.

Definition of p_t and u_t

- Let $p_t = P(x_t = 1|R = 1, \vec{q})$ be the probability of a term appearing in relevant document.
- Let $u_t = P(x_t = 1|R = 0, \vec{q})$ be the probability of a term appearing in a nonrelevant document.
- Can be displayed as contingency table:

		$R = 1$	$R = 0$
term present	$x_t = 1$	p_t	u_t
term absent	$x_t = 0$	$1 - p_t$	$1 - u_t$

Definition of p_t and u_t

- Let $p_t = P(x_t = 1|R = 1, \vec{q})$ be the probability of a term appearing in relevant document.
- Let $u_t = P(x_t = 1|R = 0, \vec{q})$ be the probability of a term appearing in a nonrelevant document.
- Can be displayed as contingency table:

		$R = 1$	$R = 0$
term present	$x_t = 1$	p_t	u_t
term absent	$x_t = 0$	$1 - p_t$	$1 - u_t$

- $$O(R|\vec{x}, \vec{q}) \propto \prod_{t:x_t=1} \frac{p_t}{u_t} \prod_{t:x_t=0} \frac{1-p_t}{1-u_t}$$

Dropping terms that don't occur in the query

Dropping terms that don't occur in the query

- Additional simplifying assumption: If $q_t = 0$, then $p_t = u_t$

Dropping terms that don't occur in the query

- Additional simplifying assumption: If $q_t = 0$, then $p_t = u_t$
  - A term not occurring in the query is equally likely to occur in relevant and nonrelevant documents.

Dropping terms that don't occur in the query

- Additional simplifying assumption: If $q_t = 0$, then $p_t = u_t$
  - A term not occurring in the query is equally likely to occur in relevant and nonrelevant documents.
- Now we need only to consider terms in the products that appear in the query:

Dropping terms that don't occur in the query

- Additional simplifying assumption: If $q_t = 0$, then $p_t = u_t$
  - A term not occurring in the query is equally likely to occur in relevant and nonrelevant documents.
- Now we need only to consider terms in the products that appear in the query:

Dropping terms that don't occur in the query

- Additional simplifying assumption: If $q_t = 0$, then $p_t = u_t$
  - A term not occurring in the query is equally likely to occur in relevant and nonrelevant documents.
- Now we need only to consider terms in the products that appear in the query:

$$O(R|\vec{x}, \vec{q}) \propto \prod_{t:x_t=1} \frac{p_t}{u_t} \prod_{t:x_t=0} \frac{1-p_t}{1-u_t} \approx \prod_{t:x_t=q_t=1} \frac{p_t}{u_t} \prod_{t:x_t=0,q_t=1} \frac{1-p_t}{1-u_t}$$

BIM retrieval status value

BIM retrieval status value

- Including the query terms found in the document into the right product, but simultaneously dividing by them in the left product, gives:

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

BIM retrieval status value

- Including the query terms found in the document into the right product, but simultaneously dividing by them in the left product, gives:

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

- The right product is now over all query terms, hence constant for a particular query and can be ignored.

BIM retrieval status value

- Including the query terms found in the document into the right product, but simultaneously dividing by them in the left product, gives:

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

- The right product is now over all query terms, hence constant for a particular query and can be ignored.
- $\rightarrow$ The only quantity that needs to be estimated to rank documents w.r.t a query is the left product.

BIM retrieval status value

- Including the query terms found in the document into the right product, but simultaneously dividing by them in the left product, gives:

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

- The right product is now over all query terms, hence constant for a particular query and can be ignored.
- $\rightarrow$ The only quantity that needs to be estimated to rank documents w.r.t a query is the left product.
- Hence the Retrieval Status Value (RSV) in this model:

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

BIM retrieval status value (2)

BIM retrieval status value (2)

Equivalent: rank documents using the log odds ratios for the terms in the query c_t:

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

- The odds ratio is the ratio of two odds: (i) the odds of the term appearing if the document is relevant ($p_t/(1-p_t)$), and (ii) the odds of the term appearing if the document is nonrelevant ($u_t/(1-u_t)$)

BIM retrieval status value (2)

Equivalent: rank documents using the log odds ratios for the terms in the query c_t:

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

- The odds ratio is the ratio of two odds: (i) the odds of the term appearing if the document is relevant ($p_t/(1-p_t)$), and (ii) the odds of the term appearing if the document is nonrelevant ($u_t/(1-u_t)$)
- $c_t = 0$: term has equal odds of appearing in relevant and nonrelevant docs

BIM retrieval status value (2)

Equivalent: rank documents using the log odds ratios for the terms in the query c_t:

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

- The odds ratio is the ratio of two odds: (i) the odds of the term appearing if the document is relevant ($p_t/(1-p_t)$), and (ii) the odds of the term appearing if the document is nonrelevant ($u_t/(1-u_t)$)
- $c_t = 0$: term has equal odds of appearing in relevant and nonrelevant docs
- c_t positive: higher odds to appear in relevant documents

BIM retrieval status value (2)

Equivalent: rank documents using the log odds ratios for the terms in the query c_t:

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

- The odds ratio is the ratio of two odds: (i) the odds of the term appearing if the document is relevant ($p_t/(1-p_t)$), and (ii) the odds of the term appearing if the document is nonrelevant ($u_t/(1-u_t)$)
- $c_t = 0$: term has equal odds of appearing in relevant and nonrelevant docs
- c_t positive: higher odds to appear in relevant documents
- c_t negative: higher odds to appear in nonrelevant documents

Term weight c_t in BIM

- $c_t = \log \frac{p_t}{(1-p_t)} - \log \frac{u_t}{1-u_t}$ functions as a term weight.

Term weight c_t in BIM

- $c_t = \log \frac{p_t}{(1-p_t)} - \log \frac{u_t}{1-u_t}$ functions as a term weight.
- Retrieval status value for document d: $RSV_d = \sum_{x_t=q_t=1} c_t$.

Term weight c_t in BIM

- $c_t = \log \frac{p_t}{(1-p_t)} - \log \frac{u_t}{1-u_t}$ functions as a term weight.
- Retrieval status value for document d: $RSV_d = \sum_{x_t=q_t=1} c_t$.
- So BIM and vector space model are similar on an operational level.

Term weight c_t in BIM

- $c_t = \log \frac{p_t}{(1-p_t)} - \log \frac{u_t}{1-u_t}$ functions as a term weight.
- Retrieval status value for document d: $RSV_d = \sum_{x_t=q_t=1} c_t$.
- So BIM and vector space model are similar on an operational level.
- In particular: we can use the same data structures (inverted index etc) for the two models.

Computing term weights c_t

For each term t in a query, estimate c_t in the whole collection using a contingency table of counts of documents in the collection, where df_t is the number of documents that contain term t:

	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

$$p_t = s/S$$

$$u_t = (\mathrm{df}_t - s)/(N - S)$$

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

Avoiding zeros

Avoiding zeros

- If any of the counts is a zero, then the term weight is not well-defined.

Avoiding zeros

- If any of the counts is a zero, then the term weight is not well-defined.
- Maximum likelihood estimates do not work for rare events.

Avoiding zeros

- If any of the counts is a zero, then the term weight is not well-defined.
- Maximum likelihood estimates do not work for rare events.
- To avoid zeros: add 0.5 to each count (expected likelihood estimation = ELE) or use a different type of smoothing

More simplifying assumptions

More simplifying assumptions

- Assume that relevant documents are a very small percentage of the collection . . .

More simplifying assumptions

- Assume that relevant documents are a very small percentage of the collection . . .
- . . . then we can approximate statistics for nonrelevant documents by statistics from the whole collection:

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

More simplifying assumptions

- Assume that relevant documents are a very small percentage of the collection ...
- ... then we can approximate statistics for nonrelevant documents by statistics from the whole collection:

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

- This should look familiar to you ...

Probability estimates in relevance feedback

Probability estimates in relevance feedback

- For relevance feedback, we can directly compute term weights c_t based on the contingency table (using an appropriate smoothing method like ELE).

Computing term weights c_t for relevance feedback

For each term t in a query, estimate c_t in the whole collection using a contingency table of counts of documents in the collection, where df_t is the number of documents that contain term t:

	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

$$p_t = s/S$$

$$u_t = (\mathrm{df}_t - s)/(N - S)$$

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

Probability estimates in adhoc retrieval

Probability estimates in adhoc retrieval

- Ad-hoc retrieval: no user-supplied relevance judgments available

Probability estimates in adhoc retrieval

- Ad-hoc retrieval: no user-supplied relevance judgments available
- In this case: assume constant $p_t = 0.5$ for all terms x_t in the query

Probability estimates in adhoc retrieval

- Ad-hoc retrieval: no user-supplied relevance judgments available
- In this case: assume constant $p_t = 0.5$ for all terms x_t in the query
- Each query term is equally likely to occur in a relevant document, and so the p_t and $(1 - p_t)$ factors cancel out in the expression for RSV.

Probability estimates in adhoc retrieval

- Ad-hoc retrieval: no user-supplied relevance judgments available
- In this case: assume constant $p_t = 0.5$ for all terms x_t in the query
- Each query term is equally likely to occur in a relevant document, and so the p_t and $(1 - p_t)$ factors cancel out in the expression for RSV.
- Weak estimate, but doesn't disagree violently with expectation that query terms appear in many but not all relevant documents.

Probability estimates in adhoc retrieval

- Ad-hoc retrieval: no user-supplied relevance judgments available
- In this case: assume constant $p_t = 0.5$ for all terms x_t in the query
- Each query term is equally likely to occur in a relevant document, and so the p_t and $(1 - p_t)$ factors cancel out in the expression for RSV.
- Weak estimate, but doesn't disagree violently with expectation that query terms appear in many but not all relevant documents.
- Weight c_t in this case: $c_t = \log \frac{p_t}{(1-p_t)} - \log \frac{u_t}{1-u_t} \approx \log N/\mathrm{df}_t$

Probability estimates in adhoc retrieval

- Ad-hoc retrieval: no user-supplied relevance judgments available
- In this case: assume constant $p_t = 0.5$ for all terms x_t in the query
- Each query term is equally likely to occur in a relevant document, and so the p_t and $(1 - p_t)$ factors cancel out in the expression for RSV.
- Weak estimate, but doesn't disagree violently with expectation that query terms appear in many but not all relevant documents.
- Weight c_t in this case: $c_t = \log \frac{p_t}{(1-p_t)} - \log \frac{u_t}{1-u_t} \approx \log N/\mathrm{df}_t$
- For short documents (titles or abstracts), this simple version of BIM works well. □

Outline

Okapi BM25: Overview

- Okapi BM25 is a probabilistic model that incorporates term frequency (i.e., it's nonbinary) and length normalization.

Okapi BM25: Overview

- Okapi BM25 is a probabilistic model that incorporates term frequency (i.e., it's nonbinary) and length normalization.
- BIM was originally designed for short catalog records of fairly consistent length, and it works reasonably in these contexts.

Okapi BM25: Overview

- Okapi BM25 is a probabilistic model that incorporates term frequency (i.e., it's nonbinary) and length normalization.
- 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.

Okapi BM25: Overview

- Okapi BM25 is a probabilistic model that incorporates term frequency (i.e., it's nonbinary) and length normalization.
- 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.
- BM25 (BestMatch25) is sensitive to these quantities. □

Okapi BM25: Starting point

Okapi BM25: Starting point

- In the simplest version of BIM, the score for document d is just idf weighting of the query terms present in the document:

Okapi BM25: Starting point

- In the simplest version of BIM, the score for document d is just idf weighting of the query terms present in the document:
- $$RSV_d = \sum_{t \in q \cap d} \log \frac{N}{\mathrm{df}_t}$$

Okapi BM25 basic weighting

Okapi BM25 basic weighting

- Improve idf term [log N/df] by factoring in term frequency and document length.

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

Okapi BM25 basic weighting

- Improve idf term [log N/df] by factoring in term frequency and document length.

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

- tf_{td}: term frequency in document d

Okapi BM25 basic weighting

- Improve idf term [log N/df] by factoring in term frequency and document length.

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

- tf_{td}: term frequency in document d
- L_d (L_{ave}): length of document d (average document length in the whole collection)

Okapi BM25 basic weighting

- Improve idf term [log N/df] by factoring in term frequency and document length.

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

- 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 scaling of term frequency

Okapi BM25 basic weighting

- Improve idf term [log N/df] by factoring in term frequency and document length.

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

- 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 scaling of term frequency
- b: tuning parameter controlling the scaling by document length

Take-away

- Probabilistic approach to IR: Introduction
- Binary independence model or BIM – the first influential probabilistic model
- Okapi BM25, a more modern, better performing probabilistic model

Resources

- Chapter 11 of Introduction to Information Retrieval
- Resources at http://informationretrieval.org/essir2011
    - Binary independence model (original paper)
    - More details on Okapi BM25
    - Why the Naive Bayes independence assumption often works (paper)

Exercise

Exercise

Naive Bayes conditional independence assumption: the presence or absence of a word in a document is independent of the presence or absence of any other word (given the query).

Exercise

Naive Bayes conditional independence assumption: the presence or absence of a word in a document is independent of the presence or absence of any other word (given the query).

Why is this wrong? Good example?

Exercise

Naive Bayes conditional independence assumption: the presence or absence of a word in a document is independent of the presence or absence of any other word (given the query).

Why is this wrong? Good example?

PRP assumes that the relevance of each document is independent of the relevance of other documents.

Exercise

Naive Bayes conditional independence assumption: the presence or absence of a word in a document is independent of the presence or absence of any other word (given the query).

Why is this wrong? Good example?

PRP assumes that the relevance of each document is independent of the relevance of other documents.

Why is this wrong? Good example?