5. Probabilistic Information Retrieval

Prof. Dr. Goran Glavaš

Data and Web Science Group
Fakultät für Wirtschaftsinformatik und Wirtschaftsmathematik
Universität Mannheim

After this lecture, you'll...

- Understand the probability ranking principle and probabilistic retrieval
- Have refreshed your knowledge of basics of probability theory
- Be familiar with the inner workings of the binary independence model (BIM)
- Learn about the more advanced probabilistic models (Two Poisson, BM25)
- Be able to compare probabilistic and vector-space ranking

Outline

- Recap of Lecture #4
- Probabilistic ranking principle
- Basics of probability theory (refresher)
- Probabilistic ranking (log-odds)
- Binary independence model (BIM)
- BIM Extensions
  - Two-Poisson model
  - BM11
  - BM25

Recap of the previous lecture

- Ranked retrieval and scoring
  - **Q:** What are issues associated with Boolean retrieval that motivate ranked retrieval?
  - **Q:** What are the common-sense assumptions of ranked retrieval?
- Vector space model
  - **Q:** What is TF-IDF weighting? How do we compute TF and how the IDF component?
  - **Q:** Can we use raw term frequency as TF component? Why (not)?
  - Q: What similarity/distance metrics do we employ in VSM?
  - **Q:** Compare cosine similarity/distance and Euclidean distance
- Optimizing VSM retrieval
  - **Q:** How can we use inverted index to speed-up VSM retrieval?
  - **Q:** What is a tiered index?
  - **Q:** How does pre-clustering work and what are *leaders*?
  - **Q:** What is locality sensitive hashing? Explain random projections

Recap of the previous lecture

- IR models for ranked retrieval
  - Produce the ordering over the documents in the collection
- Assumptions of VSM model:
  - Term more relevant the more frequent it is in the document (TF component)
  - Term more informative the less frequent it is among documents (IDF component)
- Document d_j is represented by term vector $[w_{1j}, w_{2j}, \ldots, w_{tj}]$ where each weight is the TF-IDF score of the term t_i and document d_j
- Ranking function r for VSM: cosine similarity between TF-IDF vectors of query and document
- **Today, we examine what the ranking function r looks like for the probabilistic models for ranked retrieval**
  - **Binary probabilistic model**
  - **Extensions that additionally take into account term frequency**

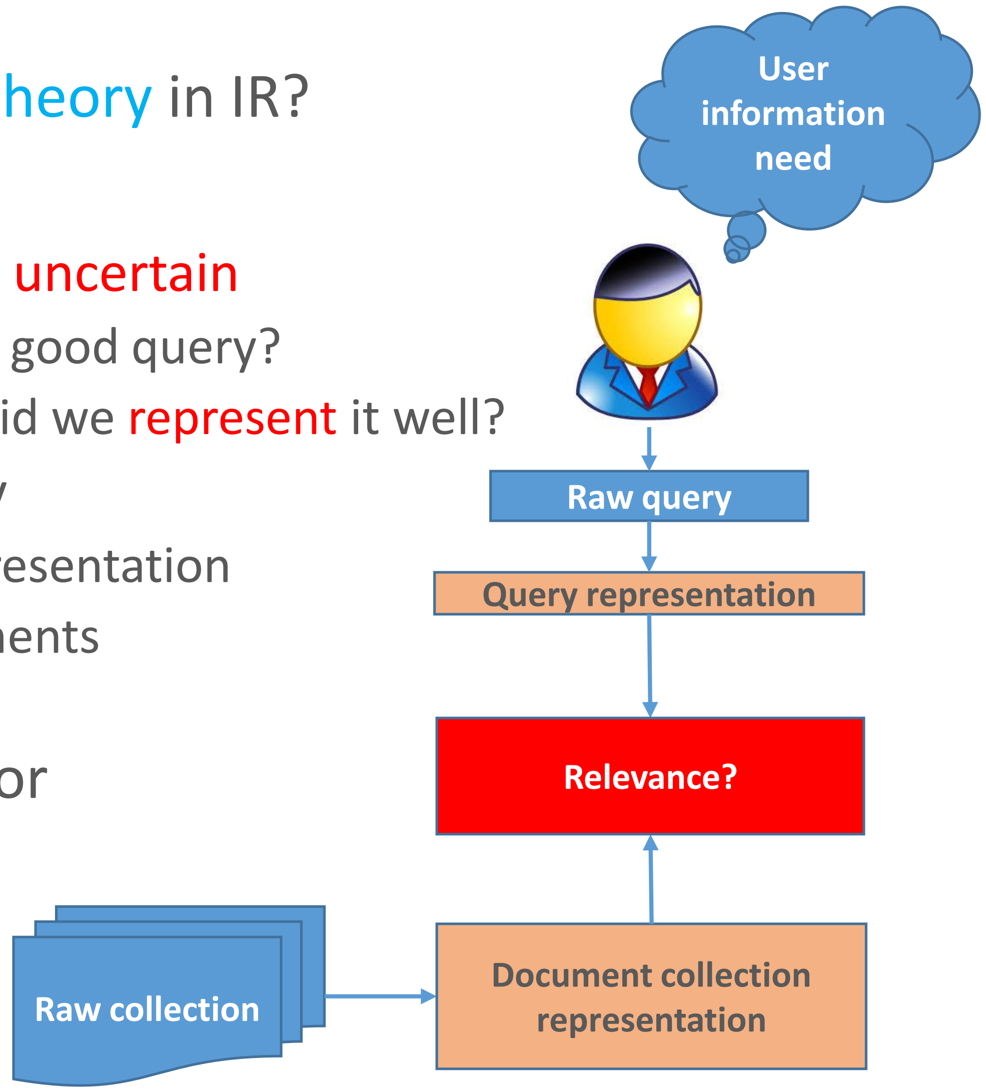
Outline

- Recap of Lecture #4
- Probabilistic ranking principle
- Basics of probability theory (refresher)
- Probabilistic ranking (log-odds)
- Binary independence model (BIM)
- BIM Extensions
  - Two-Poisson model
  - BM11
  - BM25

Probabilistic approach to retrieval

- Why introduce probabilities and probability theory in IR?
- As a process, retrieval is inherently uncertain
  - Understanding of user's information needs is uncertain
    - Are we sure the user mapped his need into a good query?
    - Even if the query represents well the need, did we represent it well?
  - Estimating document relevance for the query
    - Uncertainty from selection of document representation
    - Uncertainty from matching query and documents

- **Probability theory** is a common framework for modeling uncertainty



Probabilistic approach to retrieval

- An IR system is uncertain primarily about
  1. Understanding of the query
  2. Whether a document satisfies the query
- Probability theory
  - Provides principled foundation for reasoning under uncertainty
  - Probabilistic information retrieval models estimate how likely it is that a document is relevant for a query
- Probabilistic IR models
  - **Classic probabilistic models (BIM, Two Poisson, BM11, BM25)**
  - Language modelling for IR (next lecture)
  - Bayesian networks for text retrieval (out of scope)

- Probabilistic IR models are among the oldest, but also among the best-performing and most widely used IR models

Probabilistic ranking principle (Robertson, 1977)

- Assume the ranked retrieval setting
  - We are given a query q and a document collection D
  - Ordered list of documents from D is to be returned for q
- We model relevance (and non-relevance) as **random binary variables**
  - $R_{d,q} = 1$ if document d from D is relevant for query q,
  - $R_{d,q} = 0$ otherwise
- **Probabilistic ranking principle:** The information retrieval system will reach best obtainable efficiency if the documents are ranked decreasingly according to their probability of relevance
  - I.e., decreasingly in terms of $P(R_{d,q} = 1)$, or, equivalently, $P(R = 1 \mid d, q)$

Probabilistic ranking principle (Robertson, 1977)

Original explanation of probabilistic ranking principle (Robertson, 1977):

If [the IR] system's response to each [query] is a ranking of the documents [...] in order of decreasing probability of relevance to the [query], **where the probabilities are estimated as accurately as possible on the basis of whatever data have been made available to the system for this purpose**, the overall effectiveness of the system to its user will be the best **that is obtainable on the basis of those data**

- Probabilistic retrieval models aim to answer the following question: „What is the probability that the user will judge *this* document as relevant for *this* query?"
  - Compute the best estimate from the available data (query and document collection)
  - How do we **formalize** this question?

Formalization of the prob. ranking principle

- We introduce **sets of random variables** for terms in query and documents:

1. **$D = \{ D_1, D_2, \ldots, D_N\}$**
   - Set of random variables representing terms of documents
   - $P(D_k =$ „frodo”) -- probability of the k-th document term taking value „frodo”
2. **$Q = \{Q_1, Q_2, \ldots, Q_L\}$**
   - Set of random variables representing terms of the query
   - $P(Q_k =$ „sam”) – probability of the k-th query term taking value „sam”

- We introduce a random variable representing user’s **relevance judgement** for a concrete query-document pair

3. **$R \in \{0, 1\}$**
   - $R = 1$ if D is relevant for Q, $R = 0$ otherwise

Formalization of the prob. ranking principle

- PRP's central question:
  - „What is the probability that the user will judge *this* document as relevant for *this* query?"
- Random variables
  - $D = \{ D_1, D_2, ..., D_N\}$; $Q = \{Q_1, Q_2, ..., Q_L\}$; $R \in \{0, 1\}$
- The above question for a query q and a document d is now equivalent to estimating the probability:

$$P(R = 1 \mid D = d, Q = q)$$

- We will examine different ways of estimating this probability

Outline

- Recap of Lecture #4
- Probabilistic ranking principle
- Basics of probability theory (refresher)
- Probabilistic ranking (log-odds)
- Binary independence model (BIM)
- BIM Extensions
  - Two-Poisson model
  - BM11
  - BM25

Refresher on basics of probability theory

- Atomic events are represented with random variables – A, B, ..., Z – taking one of the possible values – a probability assigned to every event, e.g., $P(A = a)$
- For two events A and B
  - **Joint probability** $P(A, B)$ is the probability of both events ocurring
    - If events are independent, $P(A, B) = P(A) * P(B)$
  - **Conditional probability** $P(A \mid B)$ is the probability of event A occurring given the previous occurrence of event B
- **Chain rule** allows to write the joint probability using conditional probabilities

$$P(A, B, C) = P(A) * P(B \mid A) * P(C \mid A, B)$$

- **Partition rule** – if one of the events can be divided into disjoint subcases, its probability is the sum of the probabilities of the subcases

$$P(B) = P(A, B) + P(\text{not } A, B)$$

$$P(B) = P(A = a_1, B) + P(A = a_2, B) + ... + P(A = a_N, B)$$

Refresher on basics of probability theory

- **Bayes' Rule** inverts the conditional probabilities:

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

- Can be thought of as a way of updating probabilities:
  - Start off with prior probability P(A) (initial estimate of how likely event A is in the absence of any other information)
  - Derive a posterior probability P(A|B) after having seen the evidence *B*, based on the likelihood of B occurring in the two cases that *A* does or does not hold

Refresher on basics of probability theory

- **Odds** of an event occurring is a ratio of the probability of event ocurring and it not ocurring (multiplier for how probability changes)

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

- Often, instead of raw odds, we compute the logarithm of the odds, log-odds (for numeric convenience)

$$\log(O(A)) = \log P(A) – \log(1\text{-}P(A))$$

Outline

- Recap of Lecture #4
- Probabilistic ranking principle
- Basics of probability theory (refresher)
- Probabilistic ranking (log-odds)
- Binary independence model
- BIM Extensions
    - Two-Poisson model
    - BM11
    - BM25

Probabilistic relevance ranking

- We need to estimate the probability of relevance, given the query and document:

$$P(R = 1 \mid D = d, Q = q)$$

  - Let $\boldsymbol{r}$ be the shorthand for R = 1 and $\bar{r}$ be the shorthand for R = 0
- This estimate should be based on some measurable statistics that affect judgements about document's relevance for the query
  - Term frequency
  - Document frequency
  - Document length

- Ranking task formulation: order documents in decreasing order of $P(r|d,q)$
- Assumption (valid for all probabilistic models): relevance of each document is independent of the relevance of other documents for the same query (not true)

Probabilistic relevance ranking

- Ranking according to the probability P(r|D, Q) is the same as ranking according to log-odds of that probability, log(O(r|D, Q))
- Let's then start from the log-odds of the probability of relevance (simplifies math :)

$$log(O(r|D,Q)) = log\left(\frac{P(r|D,Q)}{1-P(r|D,Q)}\right)$$

$$= log\left(\frac{P(r|D,Q)}{P(\bar{r}|D,Q)}\right)$$

- Next step: apply Bayes rule on both the nominator and denominator

$$P(r|D,Q) = \frac{P(D,Q|r) \cdot P(r)}{P(D,Q)}; \qquad P(\bar{r}|D,Q) = \frac{P(D,Q|\bar{r}) \cdot P(\bar{r})}{P(D,Q)}$$

Probabilistic relevance ranking

- Because P(D, Q) cancels out, the log-odds relevance now looks like

$$log(O(r|D,Q)) = log\left(\frac{P(D,Q|r)\cdot P(r)}{P(D,Q|\bar{r})\cdot P(\bar{r})}\right)$$

- Next we use the **chain rule** to expand P(D, Q | r) and then simplify the expression by keeping only the components that depend on the document D

$$log\left(\frac{P(D,Q|r)\cdot P(r)}{P(D,Q|\bar{r})\cdot P(\bar{r})}\right) = log\left(\frac{P(D|Q,r)\cdot P(Q|r)\cdot P(r)}{P(D|Q,\bar{r})\cdot P(Q|\bar{r})\cdot P(\bar{r})}\right)$$

$$= log\left(\frac{P(D|Q,r)}{P(D|Q,\bar{r})}\right) + log\left(\frac{P(Q|r)\cdot P(r)}{P(Q|\bar{r})\cdot P(\bar{r})}\right)$$

$$\propto log\left(\frac{P(D|Q,r)}{P(D|Q,\bar{r})}\right)$$

Probabilistic relevance ranking

- The obtained probabilistic relevance is at the core of all probabilistic models:

$$log\left(\frac{P(D|Q,r)}{P(D|Q,\bar{r})}\right)$$

- We still haven't said anything on how to compute P(D|Q, *r*)
  - The way these probabilities are computed is exactly what instantiates different probabilistic retrieval models
  - In this lecture we focus on classic probabilistic retrieval models
    - Binary independence model, Two Poisson model, BM11, BM25
  - Next lecture will focus on probabilistic IR based on language modelling

Outline

- Recap of Lecture #4
- Probabilistic ranking principle
- Basics of probability theory (refresher)
- Probabilistic ranking (log-odds)
- Binary independence model
- BIM Extensions
  - Two-Poisson model
  - BM11
  - BM25

Binary independence model

- Binary independence model introduces two major assumptions that further simplify the computation of P(D|Q, *r*)

1. Independence assumption
   - Terms in the documents (and query) are independent
   - The probability of one term appearing in relevant documents does not affect the probabilities of other terms appearing in relevant documents
   - This assumption does not hold (e.g., „frodo” and „baggins”)
     - But simplifies computation and works well in practice

- Allows to represent the document probability as product of term probabilities:

$$P(D|Q,r) = \prod_{i=1}^{N} P(D_i|Q,r), \;\; P(D|Q,\bar{r}) = \prod_{i=1}^{N} P(D_i|Q,\bar{r})$$

Binary independence model

- Binary independence model introduces two major assumptions that further simplify the computation of P(D|Q, *r*)

2. Only query terms determine the relevance of the document
   - I.e., for any term D_i not in the query, probability $P(D_i|Q, r)$ does not depend on r
   - In other words, we assume:

$$P(D_i|Q,r) = P(D_i|Q,\bar{r}), \;\; log\left(\frac{P(D_i|Q,r)}{P(D_i|Q,\bar{r})}\right) = 0$$

   - Allows us to compute only the relevance probabilities for query terms:

$$P(D|Q,r) = \prod_{t \in Q} P(D_t|Q,r), \;\; P(D|Q,\bar{r}) = \prod_{t \in Q} P(D_t|Q,\bar{r})$$

Binary independence model

- Let’s integrate the binary independence model’s assumptions into the log-odds probabilistic relevance:

term independence assumption

$$log\left(\frac{P(D|Q,r)}{P(D|Q,\bar{r})}\right) = log\left(\prod_{i=1}^{N}\frac{P(D_i|Q,r)}{P(D_i|Q,\bar{r})}\right)$$

$$= \sum_{i=1}^{N} log\left(\frac{P(D_i|Q,r)}{P(D_i|Q,\bar{r})}\right)$$

assumption that only query terms affect relevance

$$= \sum_{t\in Q} log\left(\frac{P(D_t|Q,r)}{P(D_t|Q,\bar{r})}\right)$$

- Only thing left is to define how to compute the probabilities of query terms appearing in (ir)relevant documents, i.e., $P(D_t \mid Q, r)$ and $P(D_t \mid Q, \neg r)$

Binary independence model

- Based on which data/information should we compute the term-relevance probabilities $P(D_t \mid Q, r)$ and $P(D_t \mid Q, \neg r)$?

- There are two possible scenarios:
  1. We have no information which documents are relevant and which are not
     - No relevance judgements are given
  2. There are some documents which we consider relevant and/or irrelevant
     - I.e., we have a „training set", we have some relevance judgements
     - E.g., the annotations may come from (pseudo-)relevance feedback (details in Lecture 7 :)

Binary independence model

- **Scenario #1:** Estimating $P(D_t \mid Q, r)$ and $P(D_t \mid Q, \neg r)$ without relevance judgements
  - The only input information we have are query terms
  - We have no way of estimating how often query terms appear in (ir)relevant documents
- Being completely uninformed about distribution of query terms among (ir)relevant documents, we go for most reasonable assumptions
  1. Query terms equally likely to appear and not to appear in **relevant documents**

     $$P(D_t \mid Q, r) = 0.5$$

  2. Probability of the term *t* appearing in **irrelevant documents** is proportional to the number N_t of documents in the entire document collection

     $$P(D_t \mid Q, \neg r) = \frac{N_t}{N}$$

Binary independence model

- Finally, we can compute the relevance score for a document for the binary independence model without any relevance judgements:

$$rel(D,Q) = \sum_{t \in Q} log\left(\frac{P(D_t|Q,r)}{P(D_t|Q,\bar{r})}\right)$$

$$= \sum_{t \in Q} log\left(\frac{0.5}{\frac{N_t}{N}}\right)$$

$$= \sum_{t \in Q} log\left(0.5 \cdot \frac{N}{N_t}\right)$$

- The weight of each term is then: $w_t = \log(0.5 * N/N_t)$
- **Important:** w_t is computed only for terms that actually appear in the document D
- **Q:** What if $N_t = 0$?

Binary independence model – example

- Example for BIM (without relevance judgements)
- Document collection consists of the following documents:
  - d_1: „Frodo and Sam stabbed orcs”
  - d_2: „Sam chased the orc with the sword”
  - d_3: „Sam took the sword”
- The query is: „Sam stabbed orc”

	d_1			d_2		d_3
t	Sam	stabbed	orcs	Sam	orc	Sam
$P(D_t \mid q, r)$	0.5	0.5	0.5	0.5	0.5	0.5
$P(D_t \mid q, \bar{r})$	3/3	1/3	2/3	3/3	2/3	3/3
w_t	0.5	1.5	0.75	0.5	0.75	0.5
$\sum w_t$		2.75		1.25		0.5

- Note: computations in this example are done without taking the logarithm

Binary independence model

- **Scenario #2:** Estimating $P(D_t \mid Q, r)$ and $P(D_t \mid Q, \neg r)$ using relevance judgements
  - Let r_t be the number of documents judged as relevant that contain term t
  - Let R be the overall number of documents judged as relevant

- In this setting we estimate the term-relevance probabilities as follows:

$$P(D_t \mid Q, r) = \frac{r_t}{R}$$

$$P(D_t \mid Q, \neg r) = \frac{N_t - rt}{N - R}$$

- **Q:** What happens if $r_t = 0$? What happens if $r_t = N_t$?
  - **A:** we run into computational troubles (either division by 0 or log 0), we must smooth

$$P(D_t \mid Q, r) = \frac{r_t + 0.5}{R+1},\ P(D_t \mid Q, \neg r) = \frac{N_t - r_t + 0.5}{N - R + 1}$$

Binary independence model

- Finally, we can compute the relevance score for a document for the binary independence model when we have relevance judgements:

$$rel(D,Q) = \sum_{t \in Q} log\left(\frac{P(D_t|Q,r)}{P(D_t|Q,\bar{r})}\right)$$

$$= \sum_{t \in Q} log\left(\frac{\frac{r_t+0.5}{R+1}}{\frac{N_t-r_t+0.5}{N-R+1}}\right)$$

$$= \sum_{t \in Q} log\left(\frac{(r_t+0.5)\cdot(N-R+1)}{(R+1)\cdot(N_t-r_t+0.5)}\right)$$

- The weight of each term is then: $w_t = \log(\frac{(rt+0.5)(N-R+1)}{(R+1)(Nt-rt+0.5)})$
- **Important:** w_t is computed only for terms that actually appear in the document D

Binary independence model – example

- Example for BIM (with available relevance judgements)
- Document collection contains N = 30 documents, including:
  - d_1: „Frodo and Sam stabbed orcs”
  - d_2: „Sam chased the orc with the sword”
  - d_3: „Sam took the sword”
- The query is: „Sam stabbed orc”
- User has indicated $R = 6$ relevant documents for this query
- Query terms: t_1 = „Sam”, t_2 = „stab”, t_3 = „orc”
- Document frequencies of query terms in relevant documents and overall collection are given as follows:
  - $r_{t1} = 3$, $N_{t1} = 15$
  - $r_{t2} = 4$, $N_{t2} = 16$
  - $r_{t3} = 2$, $N_{t3} = 14$

Binary independence model – example

- Example for BIM (with available relevance judgements)

	d_1			d_2		d_3
t	Sam	stabbed	orcs	Sam	orc	Sam
$P(D\|Q,r) = \frac{r_t+0.5}{R+1}$	0.5	0.64	0.36	0.5	0.36	0.5
$P(D\|Q,\bar{r}) = \frac{N_t-r_t+0.5}{N-R+1}$	0.5	0.5	0.5	0.5	0.5	0.5
w_t	1	1.28	0.72	1	0.72	1
$\sum_t w_t$		3		1.72		1

- Note: computations in this example are done without taking the logarithm

Binary independence model – summary

- Probabilistic models are among the oldest formal IR models
  - An IR system cannot predict with certainty which document is relevant
  - Thus, we must deal with probabilities

- Each probablistic IR model introduces some reasonable approximations
  - In order to estimate probabilities needed for ranking

- Binary independence model employs the following approximations/assumptions:
  1. Binary (Boolean) representations of (a) documents, (b) queries, and (c) relevance
  2. Terms are considered to be mutually independent
  3. Out-of-query terms do not affect retrieval (i.e., ranking)
  4. Document relevance scores are independent (i.e., don't affect each other)

Outline

- Recap of Lecture #4
- Probabilistic ranking principle
- Basics of probability theory (refresher)
- Probabilistic ranking (log-odds)
- Binary independence model
- BIM Extensions
  - Two-Poisson model
  - BM11
  - BM25

Two Poisson model

- All BIM extensions introduce the additional scaling of term weights w_t
  - BIM: $rel(D,Q) = \sum_{t \in Q} w_t$
- We move away from binary representation – account for term frequencies
- **Two Poisson model** models frequencies with Poisson distribution
  - Implicit assumption: all documents are of equal length

$$rel(D,Q) = \sum_{t \in Q} \frac{f_{t,D}(k+1)}{f_{t,D}+k} \cdot w_t$$

  - $f_{t,D}$ is the raw frequency of term t in document D
  - k is a real constant, usually $1 \leq k < 2$
- Effect: weights of higher frequency words get boosted

BM11

- By using raw term frequency, Two Poisson model assumes all documents are equally long – but this is a faulty assumption
- **BM11 model** corrects the weight scaling factor of Two Poisson model to account for different document lengths
  - l_{avg} – average length of documents in the collection
  - l_D – the length of the document D

$$rel(D,Q) = \sum_{t \in Q} \frac{f_{t,D}(k+1)}{f_{t,D} + k\frac{l_d}{l_{avg}}} \cdot w_t$$

  - Effect: raw word frequencies dampened/boosted depending on the above/below average document length

BM25

- While BM11 removes the assumption of equal document length, in practice it has problems
  - Long relevant documents are getting too much dampening
  - Short irrelevant documents are getting too much boosting
- To control the amount of correction (dampening/boosting) for document length, the BM25 model introduces additional parameter b

$$rel(D,Q) = \sum_{t \in Q} \frac{f_{t,D}(k+1)}{f_{t,D} + k\frac{l_d}{l_{avg}}b + k(1-b)} \cdot w_t$$

  - The most common value for parameter b is b = 0.75
- BM25 is most famous probabilistic ranking function
  - Many IR systems use BM25 as the primary ranking function

Probabilistic models vs. VSM

- **Q:** Differences between probabilistic IR models and vector space model
- Different theoretical underpinnings, but similar ranking effects
  - The ranking function of the probabilistic models is grounded in probability theory
  - The ranking function of VSM – cosine similarity – is grounded in vector algebra

  - Binary independence model – binary term weights
    - Similar effect to ignoring the TF component in VSM (i.e., just IDF weighting)
  - Two-Poisson model – raw term frequency
    - Similar effect to using raw frequency as TF component in VSM
  - BM11 – accounts for document length
    - Similar effect to using length-normalized TF component in VSM ($f_{t,D}$ / max $f_{t',D}$)
  - BM25 – dampens the effects of document length
    - Similar to taking a logarithm of length-normalized frequency as TF in VSM (log($f_{t,D}$ / max $f_{t',D}$))

Now you...

- Understand the probability ranking principle and probabilistic retrieval
- Have refreshed your knowledge of basics of probability theory
- Are familiar with the inner workings of the binary independence model (BIM)
- Have learned about the more advanced probabilistic models (Two Poisson, BM25)
- Can compare probabilistic and vector-space ranking