ifis

Institut für Informationssysteme
Technische Universität Braunschweig

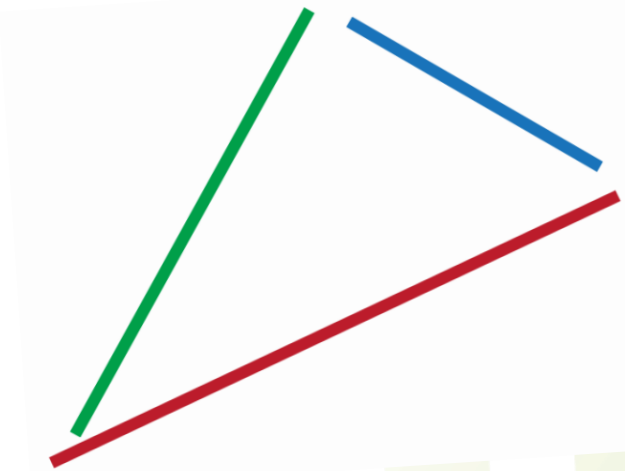
Information Retrieval and Web Search Engines

Wolf-Tilo Balke
Muhammad Usman

Institut für Informationssysteme
Technische Universität Braunschweig

Previous Lecture

1. **Fuzzy retrieval model**
2. Coordination level matching
3. Vector space retrieval model



Today's Lecture

Probabilistic Retrieval Models

1. **The Probabilistic Ranking Principle**
2. Probabilistic Indexing
3. Binary Independence Retrieval Model
4. Properties of Document Collections

Probabilistic Retrieval Models

- Probabilistic IR models use

Pr(document *d* is useful for the user asking query *q*)

as underlying **measure of similarity** between queries and documents

- Advantages:
  - Probability theory is the right tool to reason under **uncertainty** in a formal way
  - **Methods** from probability theory can be **re-used**

The Probabilistic Ranking Principle

- Probabilistic information retrieval rests upon the **Probabilistic Ranking Principle** (Robertson, 1977)

  "If a reference retrieval system's response to each request is a ranking of the documents in the collection in order of decreasing **probability of usefulness** for the user who submitted the request, where the probabilities are **estimated as accurately as possible** on the basis of whatever data has been made available to the system for this purpose, then the **overall effectiveness** of the system to its users will be **the best that is obtainable** on the basis of that data."

Usefulness and Relevance

- Characterizing **usefulness** is really tricky, we will discuss this later…
- Instead of usefulness we will consider **relevance**
- Given a **document representation** *d* and a **query** *q*, one can **objectively** determine whether *d* is relevant with respect to *q* or not
- This means in particular:
  - Relevance is a **binary** concept
  - Two documents having the same representation are either **both relevant** or **both irrelevant**

Relevance

- Denote the set of **all document representations** contained in our current collection by ***C***
- For any query *q*, denote the set of **relevant documents** contained in our collection by $\boldsymbol{R_q}$, i.e.

$$R_q = \{d \in C \mid d \text{ is relevant with respect to } q\}$$

Probability of Relevance

- **Our task then becomes:**
  - **Input:** The user's query *q* and a document *d*
  - **Output:** Pr($d \in R_q$)
- Precisely what does **this probability** mean?
  - As we have defined it,
    it is either $d \in R_q$ or $d \notin R_q$
  - Is Pr($d \in R_q$) a sensible concept?

- What does **probability in general** mean?
  - Maybe we should deal with that first…

Interpretations of Probability

Detour

- There are different interpretations of "probability," we will look at the two most common ones
- **Frequentists vs. Bayesians**
- Frequentists
  - Probability = expected **frequency** on the long run
  - Neyman, Pearson, Wald, …
- Bayesians:
  - Probability = **degree of belief**
  - Bayes, Laplace, de Finetti, …

The Frequency Interpretation Detour

- An event can be assigned a probability only if
  - It is based on **repeatable random experiments** and within this experiment, the event occurs at a persistent rate in the long run, its relative frequency.

- Examples:
  - The probability of head/tail in tossing the coin

The Bayesian Interpretation

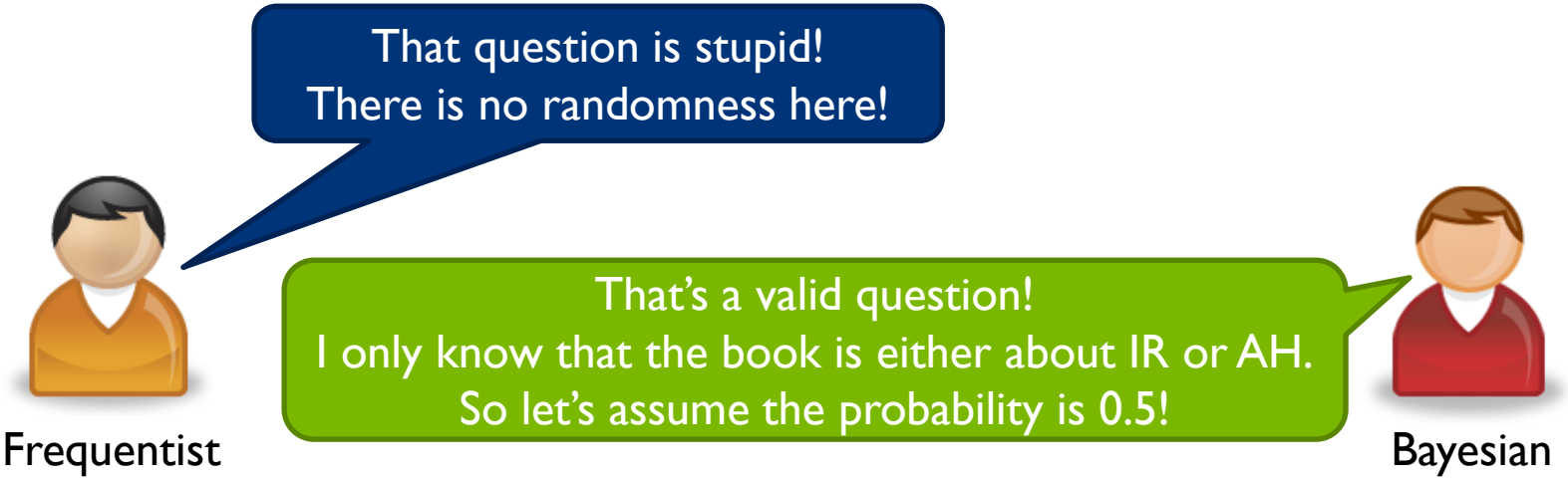
Detour

- Probability is the **degree of belief** in a proposition
- The belief can be:
  - **subjective,** i.e. personal, or
  - **objective,** i.e. justified by rational thought
- Unknown quantities are treated probabilistically
- Knowledge can always be updated
- Named after Thomas Bayes
- Examples:
  - The probability that there is life on other planets
  - The probability that **you** pass this course's exam

Frequentist vs. Bayesian

Detour

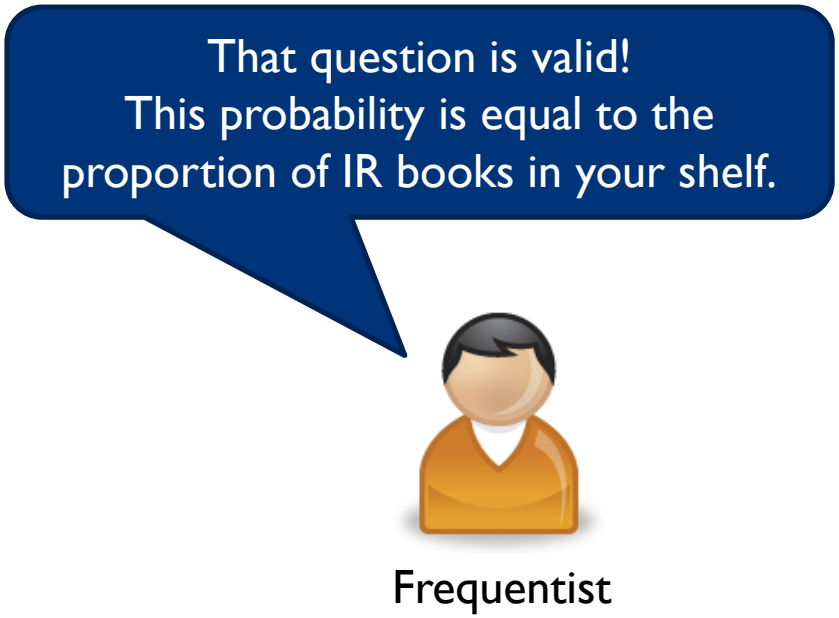
- There is a book lying on my desk
- I know it is about one of the following two topics:
  - Information retrieval
  - Animal health
- What's Pr("the book is about IR")?



Frequentist vs. Bayesian

Detour

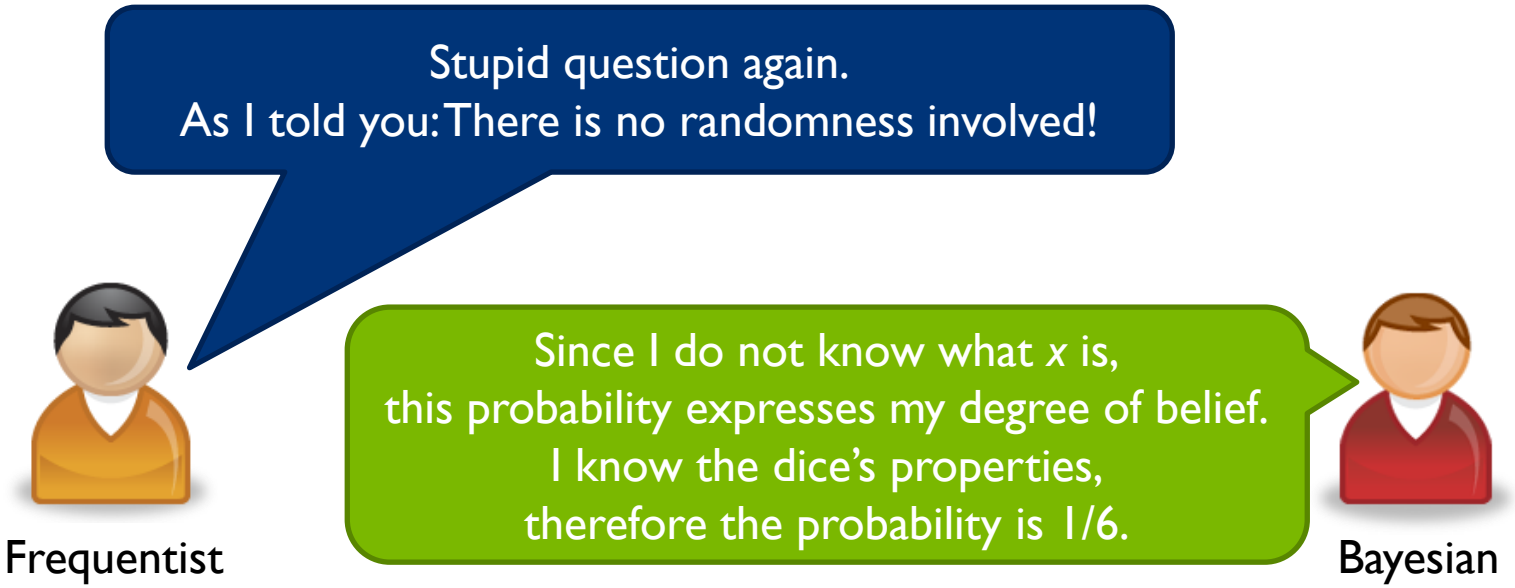
- But: Let's assume that the book is lying on my desk due to a **random draw** from my bookshelf…
- Let *X* be the "topic result" of a random draw
- What's Pr("*X* is about IR")



Frequentist vs. Bayesian

Detour

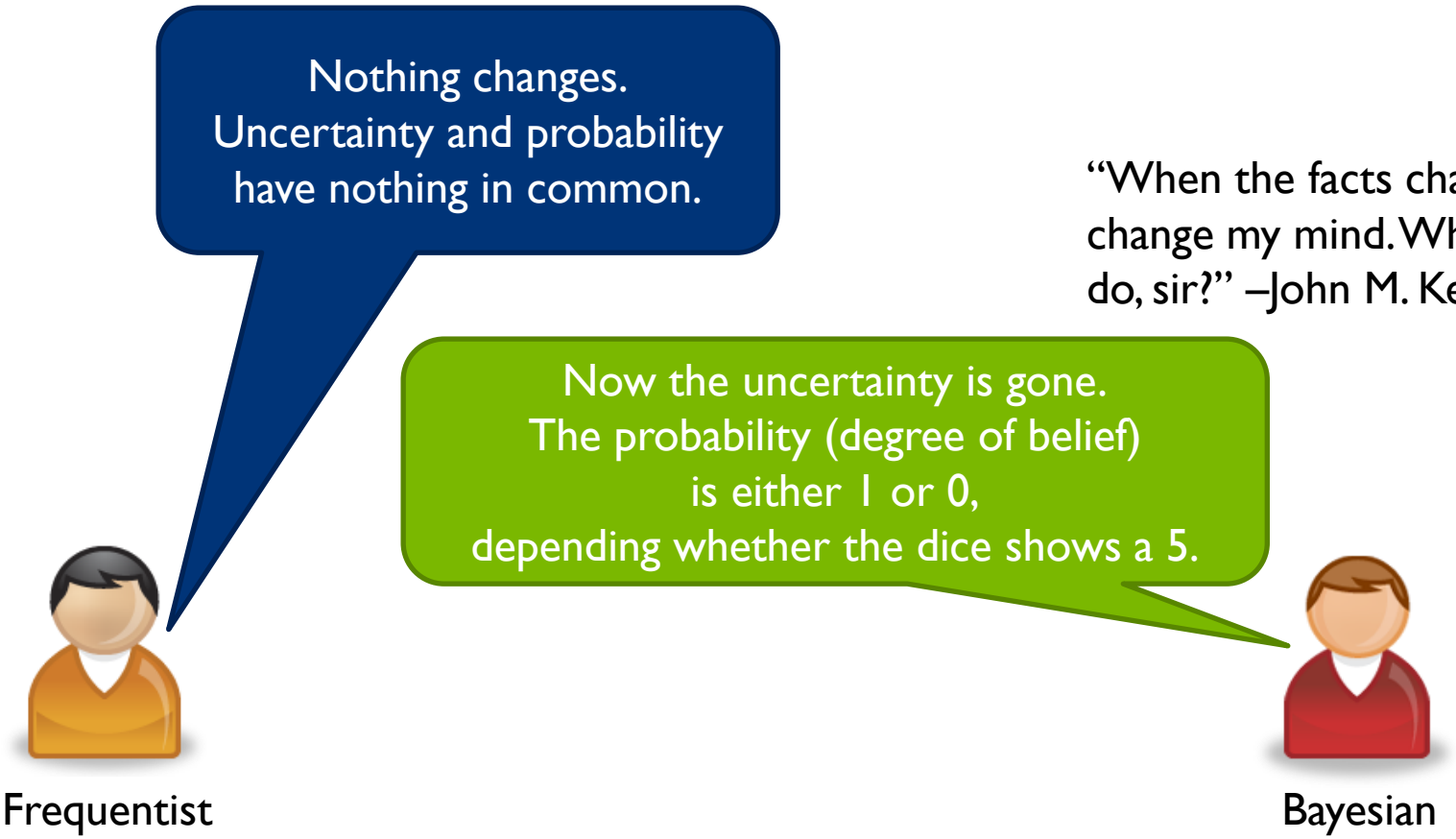
- A more practical example: Rolling a dice
- Let *x* be the (yet hidden) number on the dice that lies on the table
  Note: ***x* is a number,** not a random variable!
- What's Pr(*x* = 5)?



Frequentist vs. Bayesian

Detour

- What changes if I show you the value of *x*?



"When the facts change, I change my mind. What do you do, sir?" –John M. Keynes

Probability of Relevance, Again

- How to interpret Pr($d \in R_q$)?
  - Clearly: **Bayesian** (expressing uncertainty regarding R_q)
- Although there is a crisp set R_q (by assumption), **we do not know what R_q looks like**
- Bayesian approach:
  Express uncertainty in terms of probability
- Probabilistic models of information retrieval:
  - Start with Pr($d \in R_q$) and **relate it to other probabilities,** which might be more easily accessible
  - On this way, make some **reasonable assumptions**
  - Finally, **estimate Pr($d \in R_q$)** using other probabilities' estimates

Today's Lecture

Probabilistic Retrieval Models

1. The Probabilistic Ranking Principle
2. **Probabilistic Indexing**
3. Binary Independence Retrieval Model
4. Properties of Document Collections

Probabilistic Indexing

- Presented by Maron and Kuhns in 1960
- Goal: Improve automatic search on **manually indexed** document collections
- **Basic notions:**
  - k index terms
  - Documents = vectors over $[0, 1]^k$, i.e. **terms are weighted**
  - Queries = vectors over $\{0, 1\}^k$, i.e. **binary queries**
  - R_q = relevant documents with respect to query q (as above)
- **Task:**
  Given a query q, estimate $\Pr(d \in R_q)$, for each document d

Probabilistic Indexing

- Let *Q* be a **random variable** ranging over the set of all possible queries
- *Q*'s distribution corresponds to the sequence of all queries asked in the past
  - Example ($k = 2$):
    Ten queries have been asked to the system previously:

(0, 0)	(1, 0)	(0, 1)	(1, 1)
0	2	7	1

Then, *Q*'s distribution is given by:

Pr(Q = (0, 0))	Pr(Q = (1, 0))	Pr(Q = (0, 1))	Pr(Q = (1, 1))
0	0.2	0.7	0.1

Probabilistic Indexing

- If Q is a **random query,**
  then R_Q is a **random set of documents**
- We can use R_Q to express
  our initial probability $\Pr(d \in R_q)$:

$$\Pr(d \in R_q) = \Pr(d \in R_Q \mid Q = q)$$

- This means:
  If we restrict our view to events where Q is equal to q,
  then $\Pr(d \in R_q)$ is equal to $\Pr(d \in R_Q)$

Probabilistic Indexing

$$\Pr(d \in R_q) = \Pr(d \in R_Q \mid Q = q)$$

- Now, let's apply **Bayes' Theorem:**

$$\Pr(d \in R_Q \mid Q = q) = \frac{\Pr(d \in R_Q)}{\Pr(Q = q)} \cdot \Pr(Q = q \mid d \in R_Q)$$

- Combined:

$$\Pr(d \in R_q) = \frac{\Pr(d \in R_Q)}{\Pr(Q = q)} \cdot \Pr(Q = q \mid d \in R_Q)$$

Probabilistic Indexing

$$\Pr(d \in R_q) = \frac{\Pr(d \in R_Q)}{\Pr(Q = q)} \cdot \Pr(Q = q \mid d \in R_Q)$$

- $\Pr(Q = q)$ is the same for all documents d
- Therefore, the document **ranking** induced by $\Pr(d \in R_q)$ is identical to the ranking induced by $\Pr(d \in R_Q) \cdot \Pr(Q = q \mid d \in R_Q)$
- Since we are only interested in the ranking, we can replace $\Pr(Q = q)$ by a **constant:**

$$\Pr(d \in R_q) = c(q) \cdot \Pr(d \in R_Q) \cdot \Pr(Q = q \mid d \in R_Q)$$

Probabilistic Indexing

$$\Pr(d \in R_q) = c(q) \cdot \Pr(d \in R_Q) \cdot \Pr(Q = q \mid d \in R_Q)$$

- $\Pr(d \in R_Q)$ can be estimated from **user feedback**
  - Give the users a mechanism to **rate** whether the document they read previously has been **relevant** with respect to their query
  - $\Pr(d \in R_Q)$ is the relative frequency of positive relevance ratings

- Finally, we must estimate $\Pr(Q = q \mid d \in R_Q)$

Probabilistic Indexing

- How to estimate $\Pr(Q = q \mid d \in R_Q)$?
- Assume **independence of query terms:**

$$\Pr(Q = q \mid d \in R_Q) = \prod_{i=1}^{k} \Pr(Q_i = q_i \mid d \in R_Q)$$

- Is this assumption reasonable?
  - Obviously not (co-occurrence, think of synonyms)!

Probabilistic Indexing

$$\Pr(Q = q \mid d \in R_Q) = \prod_{i=1}^{k} \Pr(Q_i = q_i \mid d \in R_Q)$$

- What's next? Split up the product by q_i's value!

$$\Pr(Q = q \mid d \in R_Q) = \prod_{\substack{i \in \{1,\ldots,k\} \\ q_i = 0}} \Pr(Q_i = 0 \mid d \in R_Q) \cdot \prod_{\substack{i \in \{1,\ldots,k\}, \\ q_i = 1}} \Pr(Q_i = 1 \mid d \in R_Q)$$

- Look at complementary events:

$$\Pr(Q = q \mid d \in R_Q)$$
$$= \prod_{\substack{i \in \{1,\ldots,k\} \\ q_i = 0}} \left(1 - \Pr(Q_i = 1 \mid d \in R_Q)\right) \cdot \prod_{\substack{i \in \{1,\ldots,k\}, \\ q_i = 1}} \Pr(Q_i = 1 \mid d \in R_Q)$$

Probabilistic Indexing

$$\Pr(Q = q \mid d \in R_Q)$$
$$= \prod_{\substack{i \in \{1,\ldots,k\} \\ q_i = 0}} \left(1 - \Pr(Q_i = 1 \mid d \in R_Q)\right) \cdot \prod_{\substack{i \in \{1,\ldots,k\}, \\ q_i = 1}} \Pr(Q_i = 1 \mid d \in R_Q)$$

- Only $\Pr(Q_i = 1 \mid d \in R_Q)$ remains unknown
- It corresponds to the following:
  Given that document *d* is relevant for some query,
  what is the probability that the query contained term *i*?

Probabilistic Indexing

$$\Pr(Q_i = 1 \mid d \in R_Q)$$

- Given that document *d* is relevant for some query, what is the probability that the query contained term *i*?
- Maron and Kuhns argue that $\Pr(Q_i = 1 \mid d \in R_Q)$ can be estimated by the **weight of term *i*** assigned to *d* by the human indexer
- Is this assumption reasonable? Yes!
  1. The indexer knows that the current document to be indexed definitely is relevant with respect to some topics
  2. She/he then tries to find out what these topics are
     - Topics correspond to index terms
     - Term weights represent degrees of belief

Probabilistic Indexing

- Taken all together, we arrive at:

$$\Pr(d \in R_q) = c(q) \cdot \Pr(d \in R_Q) \cdot \prod_{\substack{i \in \{1,\ldots,k\} \\ q_i = 0}} (1 - d_i) \cdot \prod_{\substack{i \in \{1,\ldots,k\}, \\ q_i = 1}} d_i$$

- $c(q)$ doesn't matter
- $\Pr(d \in R_Q)$ can be estimated from query logs

- Possible modification:
  - Remove the $(1 - d_i)$ factors, since most users leave out query terms unintentionally

Reality Check

$$\Pr(d \in R_q) = c(q) \cdot \Pr(d \in R_Q) \cdot \prod_{\substack{i \in \{1,\ldots,k\} \\ q_i = 0}} (1 - d_i) \cdot \prod_{\substack{i \in \{1,\ldots,k\}, \\ q_i = 1}} d_i$$

- $\Pr(d \in R_Q)$ models the "general relevance" of d
  - $\Pr(d \in R_q)$ is proportional to $\Pr(d \in R_Q)$
  - This is reasonable
  - Think of the following example:
    - You want to buy a book at a book store
    - Book *A*'s description **almost perfectly** fits what you are looking for
    - Book B's description **perfectly** fits what you are looking for
    - Book *A* is a bestseller
    - Nobody else is interested in book *B*
    - Which book is better?

Today's Lecture

Probabilistic Retrieval Models

1. The Probabilistic Ranking Principle
2. Probabilistic Indexing
3. **Binary Independence Retrieval Model**
4. Properties of Document Collections

Binary Independence Retrieval

- Presented by van Rijsbergen in 1977
- **Basic notions:**
  - k index terms
  - Documents = vectors over $\{0, 1\}^k$, i.e. **set of words** model
  - Queries = vectors over $\{0, 1\}^k$, i.e. **set of words** model
  - R_q = relevant documents with respect to query q
- **Task:**
  Given a query q, estimate $\Pr(d \in R_q)$, for any document d

Binary Independence Retrieval

- Let D be a **uniformly** distributed **random variable** ranging over the set of all documents in the collection
- We can use D to express our initial probability $\Pr(d \in R_q)$:

$$\Pr(d \in R_q) = \Pr(D \in R_q \mid D = d)$$

- This means:
  If we restrict our view to events where D is equal to d, then $\Pr(d \in R_q)$ is equal to $\Pr(D \in R_q)$
- Note the **similarity to probabilistic indexing:**

$$\Pr(d \in R_q) = \Pr(d \in R_Q \mid Q = q)$$

Binary Independence Retrieval

$$\Pr(d \in R_q) = \Pr(D \in R_q \mid D = d)$$

- Again, let's apply **Bayes' Theorem:**

$$\Pr(D \in R_q \mid D = d) = \frac{\Pr(D \in R_q)}{\Pr(D = d)} \cdot \Pr(D = d \mid D \in R_q)$$

- Combined:

$$\Pr(d \in R_q) = \frac{\Pr(D \in R_q)}{\Pr(D = d)} \cdot \Pr(D = d \mid D \in R_q)$$

Binary Independence Retrieval

$$\Pr(d \in R_q) = \frac{\Pr(D \in R_q)}{\Pr(D = d)} \cdot \Pr(D = d \mid D \in R_q)$$

- $\Pr(D \in R_q)$ is identical for all documents d
- Since we are only interested in the probability ranking, we can replace $\Pr(D \in R_q)$ by a constant:

$$\Pr(d \in R_q) = c(q) \cdot \frac{1}{\Pr(D = d)} \cdot \Pr(D = d \mid D \in R_q)$$

Binary Independence Retrieval

$$\Pr(d \in R_q) = c(q) \cdot \frac{1}{\Pr(D = d)} \cdot \Pr(D = d \mid D \in R_q)$$

- $\Pr(D = d)$ represents the proportion of documents in the collection having the same representation as d
- Although we know this probability, it basically is an **artifact** of our approach to transforming $\Pr(d \in R_q)$ into something Bayes' Theorem can be applied on
- Unconditionally reducing highly popular documents in rank simply makes no sense

Binary Independence Retrieval

$$\Pr(d \in R_q) = c(q) \cdot \frac{1}{\Pr(D = d)} \cdot \Pr(D = d \mid D \in R_q)$$

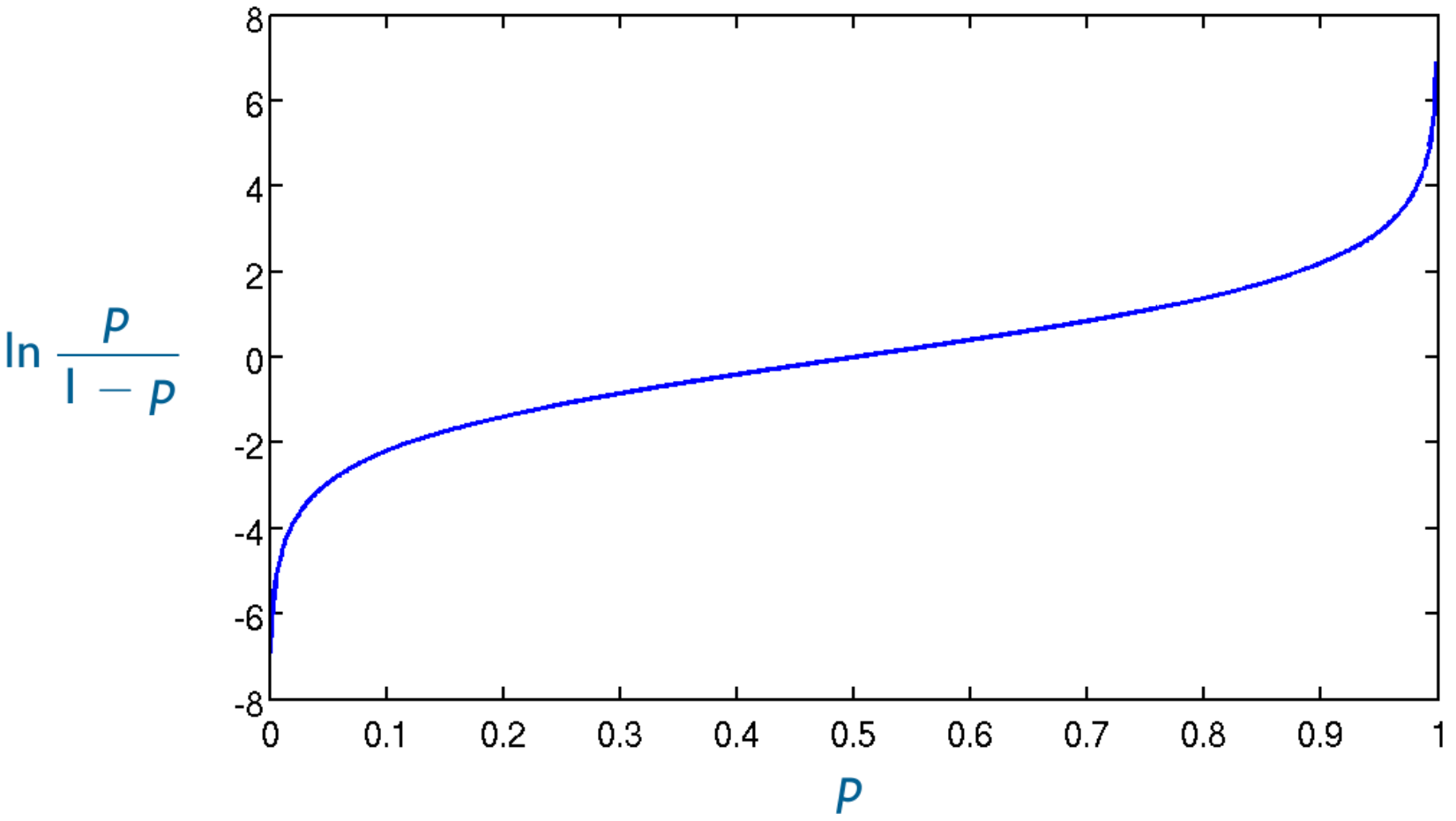
- How to get rid of $\Pr(D = d)$?
- Instead of $\Pr(d \in R_q)$ we look at its **odds:**

$$\text{Odds}(d \in R_q) = \frac{\Pr(d \in R_q)}{1 - \Pr(d \in R_q)} = \frac{\Pr(d \in R_q)}{\Pr(d \notin R_q)}$$

- As we will see on the next slide, ordering documents by this odds results in the **same ranking** as ordering by probability

Binary Independence Retrieval

- This graph depicts **probability versus (log) odds:**



Binary Independence Retrieval

$$\text{Odds}(d \in R_q) = \frac{\Pr(d \in R_q)}{\Pr(d \notin R_q)}$$

- Applying **Bayes' Theorem** on $\Pr(d \notin R_q)$ yields:

$$\Pr(d \notin R_q) = c(q) \cdot \frac{1}{\Pr(D = d)} \cdot \Pr(D = d \mid D \notin R_q)$$

- Again, $c(q)$ is a constant that is independent of d

Binary Independence Retrieval

$$\text{Odds}(d \in R_q) = \frac{\Pr(d \in R_q)}{\Pr(d \notin R_q)}$$

$$\Pr(d \in R_q) = c(q) \cdot \frac{1}{\Pr(D = d)} \cdot \Pr(D = d \mid D \in R_q)$$

$$\Pr(d \notin R_q) = c(q) \cdot \frac{1}{\Pr(D = d)} \cdot \Pr(D = d \mid D \notin R_q)$$

- Putting it all together we arrive at:

$$\text{Odds}(d \in R_q) = c(q) \cdot \frac{\Pr(D = d \mid D \in R_q)}{\Pr(D = d \mid D \notin R_q)}$$

Binary Independence Retrieval

$$\text{Odds}(d \in R_q) = c(q) \cdot \frac{\Pr(D = d \mid D \in R_q)}{\Pr(D = d \mid D \notin R_q)}$$

- It looks like we need an **assumption**
- Assumption of **linked dependence:**

$$\frac{\Pr(D = d \mid D \in R_q)}{\Pr(D = d \mid D \notin R_q)} = \prod_{i=1}^{k} \frac{\Pr(D_i = d_i \mid D \in R_q)}{\Pr(D_i = d_i \mid D \notin R_q)}$$

(slightly weaker than assuming independent terms)

- Is this assumption reasonable?
  - No, think of synonyms…

Binary Independence Retrieval

$$\text{Odds}(d \in R_q) = c(q) \cdot \prod_{i=1}^{k} \frac{\Pr(D_i = d_i \mid D \in R_q)}{\Pr(D_i = d_i \mid D \notin R_q)}$$

- Let's **split it up** by term occurrences within *d*:

$$\text{Odds}(d \in R_q) = c(q) \cdot \prod_{\substack{i \in \{1,\ldots,k\} \\ d_i = 0}} \frac{\Pr(D_i = 0 \mid D \in R_q)}{\Pr(D_i = 0 \mid D \notin R_q)} \cdot \prod_{\substack{i \in \{1,\ldots,k\} \\ d_i = 1}} \frac{\Pr(D_i = 1 \mid D \in R_q)}{\Pr(D_i = 1 \mid D \notin R_q)}$$

- Replace $\Pr(D_i = 0 \mid \ldots)$ by $1 - \Pr(D_i = 1 \mid \ldots)$:

$$\text{Odds}(d \in R_q) = c(q) \cdot \prod_{\substack{i \in \{1,\ldots,k\} \\ d_i = 0}} \frac{1 - \Pr(D_i = 1 \mid D \in R_q)}{1 - \Pr(D_i = 1 \mid D \notin R_q)} \cdot \prod_{\substack{i \in \{1,\ldots,k\} \\ d_i = 1}} \frac{\Pr(D_i = 1 \mid D \in R_q)}{\Pr(D_i = 1 \mid D \notin R_q)}$$

Binary Independence Retrieval

$$\text{Odds}(d \in R_q) = c(q) \cdot \prod_{\substack{i \in \{1,\ldots,k\} \\ d_i = 0}} \frac{1 - \Pr(D_i = 1 \mid D \in R_q)}{1 - \Pr(D_i = 1 \mid D \notin R_q)} \cdot \prod_{\substack{i \in \{1,\ldots,k\} \\ d_i = 1}} \frac{\Pr(D_i = 1 \mid D \in R_q)}{\Pr(D_i = 1 \mid D \notin R_q)}$$

- Let's **split it up** by term occurrences within q:

$$\text{Odds}(d \in R_q) = c(q) \cdot \prod_{\substack{i \in \{1,\ldots,k\}, \\ d_i = 0, \\ q_i = 0}} \frac{1 - \Pr(D_i = 1 \mid D \in R_q)}{1 - \Pr(D_i = 1 \mid D \notin R_q)} \cdot \prod_{\substack{i \in \{1,\ldots,k\}, \\ d_i = 0, \\ q_i = 1,}} \frac{1 - \Pr(D_i = 1 \mid D \in R_q)}{1 - \Pr(D_i = 1 \mid D \notin R_q)}$$

$$\cdot \prod_{\substack{i \in \{1,\ldots,k\}, \\ d_i = 1, \\ q_i = 0}} \frac{\Pr(D_i = 1 \mid D \in R_q)}{\Pr(D_i = 1 \mid D \notin R_q)} \cdot \prod_{\substack{i \in \{1,\ldots,k\}, \\ d_i = 1, \\ q_i = 1}} \frac{\Pr(D_i = 1 \mid D \in R_q)}{\Pr(D_i = 1 \mid D \notin R_q)}$$

Binary Independence Retrieval

$$\text{Odds}(d \in R_q) = c(q) \cdot \prod_{\substack{i \in \{1,\ldots,k\},\\ d_i=0,\\ q_i=0}} \frac{1 - \Pr(D_i = 1 \mid D \in R_q)}{1 - \Pr(D_i = 1 \mid D \notin R_q)} \cdot \prod_{\substack{i \in \{1,\ldots,k\},\\ d_i=0,\\ q_i=1,}} \frac{1 - \Pr(D_i = 1 \mid D \in R_q)}{1 - \Pr(D_i = 1 \mid D \notin R_q)}$$

$$\cdot \prod_{\substack{i \in \{1,\ldots,k\},\\ d_i=1,\\ q_i=0}} \frac{\Pr(D_i = 1 \mid D \in R_q)}{\Pr(D_i = 1 \mid D \notin R_q)} \cdot \prod_{\substack{i \in \{1,\ldots,k\},\\ d_i=1,\\ q_i=1}} \frac{\Pr(D_i = 1 \mid D \in R_q)}{\Pr(D_i = 1 \mid D \notin R_q)}$$

- Looks like we heavily need an **assumption…**
- Assume that $\Pr(D_i = 1 \mid D \in R_q) = \Pr(D_i = 1 \mid D \notin R_q)$, for any i such that $q_i = 0$
- **Idea:** Relevant and non-relevant documents have **identical term distributions for non-query terms**
- Consequence: Two of the four product blocks cancel out

Binary Independence Retrieval

- This leads us to:

$$\text{Odds}(d \in R_q) = c(q) \cdot \prod_{\substack{i \in \{1,\ldots,k\},\\ d_i=0,\\ q_i=1,}} \frac{1 - \Pr(D_i = 1 \mid D \in R_q)}{1 - \Pr(D_i = 1 \mid D \notin R_q)} \cdot \prod_{\substack{i \in \{1,\ldots,k\},\\ d_i=1,\\ q_i=1}} \frac{\Pr(D_i = 1 \mid D \in R_q)}{\Pr(D_i = 1 \mid D \notin R_q)}$$

- Multiply by 1 and regroup:

$$1 = \prod_{\substack{i \in \{1,\ldots,k\},\\ d_i=1,\\ q_i=1,}} \frac{1 - \Pr(D_i = 1 \mid D \in R_q)}{1 - \Pr(D_i = 1 \mid D \notin R_q)} \cdot \frac{1 - \Pr(D_i = 1 \mid D \notin R_q)}{1 - \Pr(D_i = 1 \mid D \in R_q)}$$

$$\text{Odds}(d \in R_q)$$

$$= c(q) \cdot \prod_{\substack{i \in \{1,\ldots,k\},\\ q_i=1}} \frac{1 - \Pr(D_i = 1 \mid D \in R_q)}{1 - \Pr(D_i = 1 \mid D \notin R_q)} \cdot \prod_{\substack{i \in \{1,\ldots,k\},\\ d_i=1,\\ q_i=1}} \frac{\Pr(D_i = 1 \mid D \in R_q) \cdot \left(1 - \Pr(D_i = 1 \mid D \notin R_q)\right)}{\Pr(D_i = 1 \mid D \notin R_q) \cdot \left(1 - \Pr(D_i = 1 \mid D \in R_q)\right)}$$

Binary Independence Retrieval

$$\text{Odds}(d \in R_q)$$

$$= c(q) \cdot \prod_{\substack{i \in \{1,\ldots,k\},\\ q_i = 1}} \frac{1 - \Pr(D_i = 1 \mid D \in R_q)}{1 - \Pr(D_i = 1 \mid D \notin R_q)} \cdot \prod_{\substack{i \in \{1,\ldots,k\},\\ d_i = 1,\\ q_i = 1}} \frac{\Pr(D_i = 1 \mid D \in R_q) \cdot \left(1 - \Pr(D_i = 1 \mid D \notin R_q)\right)}{\Pr(D_i = 1 \mid D \notin R_q) \cdot \left(1 - \Pr(D_i = 1 \mid D \in R_q)\right)}$$

- Fortunately, the first product block is independent of d, so we can replace it by a constant:

$$\text{Odds}(d \in R_q) = c(q) \cdot \prod_{\substack{i \in \{1,\ldots,k\},\\ d_i = 1,\\ q_i = 1}} \frac{\Pr(D_i = 1 \mid D \in R_q) \cdot \left(1 - \Pr(D_i = 1 \mid D \notin R_q)\right)}{\Pr(D_i = 1 \mid D \notin R_q) \cdot \left(1 - \Pr(D_i = 1 \mid D \in R_q)\right)}$$

$$= c(q) \cdot \prod_{\substack{i \in \{1,\ldots,k\},\\ d_i = 1,\\ q_i = 1}} \left(\frac{\Pr(D_i = 1 \mid D \in R_q)}{1 - \Pr(D_i = 1 \mid D \in R_q)} \cdot \frac{1 - \Pr(D_i = 1 \mid D \notin R_q)}{\Pr(D_i = 1 \mid D \notin R_q)} \right)$$

Binary Independence Retrieval

$$\text{Odds}(d \in R_q) = c(q) \cdot \prod_{\substack{i \in \{1,\ldots,k\},\\ d_i = 1,\\ q_i = 1}} \left(\frac{\Pr(D_i = 1 \mid D \in R_q)}{1 - \Pr(D_i = 1 \mid D \in R_q)} \cdot \frac{1 - \Pr(D_i = 1 \mid D \notin R_q)}{\Pr(D_i = 1 \mid D \notin R_q)} \right)$$

- How to estimate the second quotient?
- Since usually most documents in the collection will not be relevant to q, we can assume the following:

$$\Pr(D_i = 1 \mid D \notin R_q) \approx \Pr(D_i = 1)$$

- Reasonable assumption?

Binary Independence Retrieval

$$\Pr(D_i = 1 \mid D \notin R_q) \approx \Pr(D_i = 1)$$

- How to estimate $\Pr(D_i = 1)$?
- $\Pr(D_i = 1)$ is roughly the proportion of documents in the collection containing term *i*:

$$\Pr(D_i = 1) \approx \frac{\mathrm{df}(t_i)}{N}$$

- *N*: collection size
- $\mathrm{df}(t_i)$: document frequency of term *i*

Binary Independence Retrieval

$$\Pr(D_i = 1) \approx \frac{\mathrm{df}(t_i)}{N}$$

- This leads us to the final estimate:

$$\frac{1 - \Pr(D_i = 1 \mid D \notin R_q)}{\Pr(D_i = 1 \mid D \notin R_q)} \approx \frac{1 - \frac{\mathrm{df}(t_i)}{N}}{\frac{\mathrm{df}(t_i)}{N}} = \frac{N - \mathrm{df}(t_i)}{\mathrm{df}(t_i)} \approx \frac{N}{\mathrm{df}(t_i)}$$

Binary Independence Retrieval (19)

$$\text{Odds}(d \in R_q) \approx c(q) \cdot \prod_{\substack{i \in \{1,\ldots,k\}, \\ d_i = 1, \\ q_i = 1}} \left(\frac{\Pr(D_i = 1 \mid D \in R_q)}{1 - \Pr(D_i = 1 \mid D \in R_q)} \cdot \frac{N}{\text{df}(t_i)} \right)$$

- $\Pr(D_i = 1 \mid D \in R_q)$ cannot be estimated that easy…
- There are several options:
  - Estimate it from **user feedback** on initial result lists
  - Estimate it by a **constant** (Croft and Harper, 1979), e.g. 0.9
  - Estimate it by **$\text{df}(t_i) / N$** (Greiff, 1998)

Probabilistic Models

- Are there any other probabilistic models?
- Of course:
  - Extension of the Binary Independence Retrieval model
    - Learning from user feedback
    - Different types of queries
    - Accounting for dependencies between terms
  - Poisson model
  - Belief networks
  - Many more…

Pros and Cons

- **Pros**
  - Very successful in experiments
  - Probability of relevance as intuitive measure
  - Well-developed mathematical foundations
  - All assumptions can be made explicit

- **Cons**
  - Estimation of parameters usually is difficult
  - Doubtful assumptions
  - Much less flexible than the vector space model
  - Quite complicated

Today's Lecture

Probabilistic Retrieval Models

1. The Probabilistic Ranking Principle
2. Probabilistic Indexing
3. Binary Independence Retrieval Model
4. **Properties of Document Collections**

Some Statistics

Detour

- Some data for two test collections:

	Newswire	Web
Size	1 GB	100 GB
Documents	400,000	12,000,000
Posting entries	180,000,000	11,000,000,000
Vocabulary size (after stemming)	400,000	16,000,000
Index size (uncompressed, without word positions)	450 MB	21 GB
Index size (uncompressed, with word positions)	800 MB	43 GB
Index size (compressed, with word positions)	130 MB	?

Source: (Zobel and Moffat, 2006)

Heaps' Law

Detour

- How big is the term vocabulary?
- Clearly, there must be an upper bound
  - The num… "… words
  - Whe… …tion g…
    the vocabulary size will converge to this number
- **Sorry, this is simply wrong…**

Heaps' law: $\#\text{terms} = k \cdot (\#\text{tokens})^b$

- k and b are positive constants, collection-dependent
- Typical values: $30 \leq k \leq 100$, $b \approx 0.5$
- **Empirically verified** for many different collections

Heaps' Law

Detour

Heaps' law: #terms = $k \cdot (\#\text{tokens})^b$

- Example:
  - Looking at a collection of web pages, you find that there are 3,000 different terms in the first 10,000 tokens and 30,000 different terms in the first 1,000,000 tokens
  - Assume a search engine indexes a total of 20,000,000,000 pages, containing 200 tokens on average
  - What is the size of the vocabulary of the indexed collection as predicted by Heaps' law?

$$3{,}000 = k \cdot 10{,}000^b \qquad 30{,}000 = k \cdot 1{,}000{,}000^b$$

$$\Rightarrow k = 30,\ b = 0.5$$

$\Rightarrow$ The vocabulary size is $30 \cdot 4{,}000{,}000{,}000{,}000^{0.5} = 60{,}000{,}000$

Zipf's Law

Detour

Zipf's law: The *i*-th most frequent term has frequency proportional to 1 / *i*

- **Key insights:**
  - Few frequent terms
  - Many rare terms

  } **A heavily skewed data distribution! That's why compression of posting lists works so well in practice!**

- Zipf's law is an example of a **power law:**

$$\mathbf{Pr}(x) = a \cdot x^b$$

  - *a* is a normalization constant (total probability mass must be 1)
  - **In Zipf's law: $b \approx -1$**

Zipf's Law

Detour

- Zipf analyzed samples of natural language
  - Letter frequencies
  - Term frequencies

- **Letter frequencies** in English language:

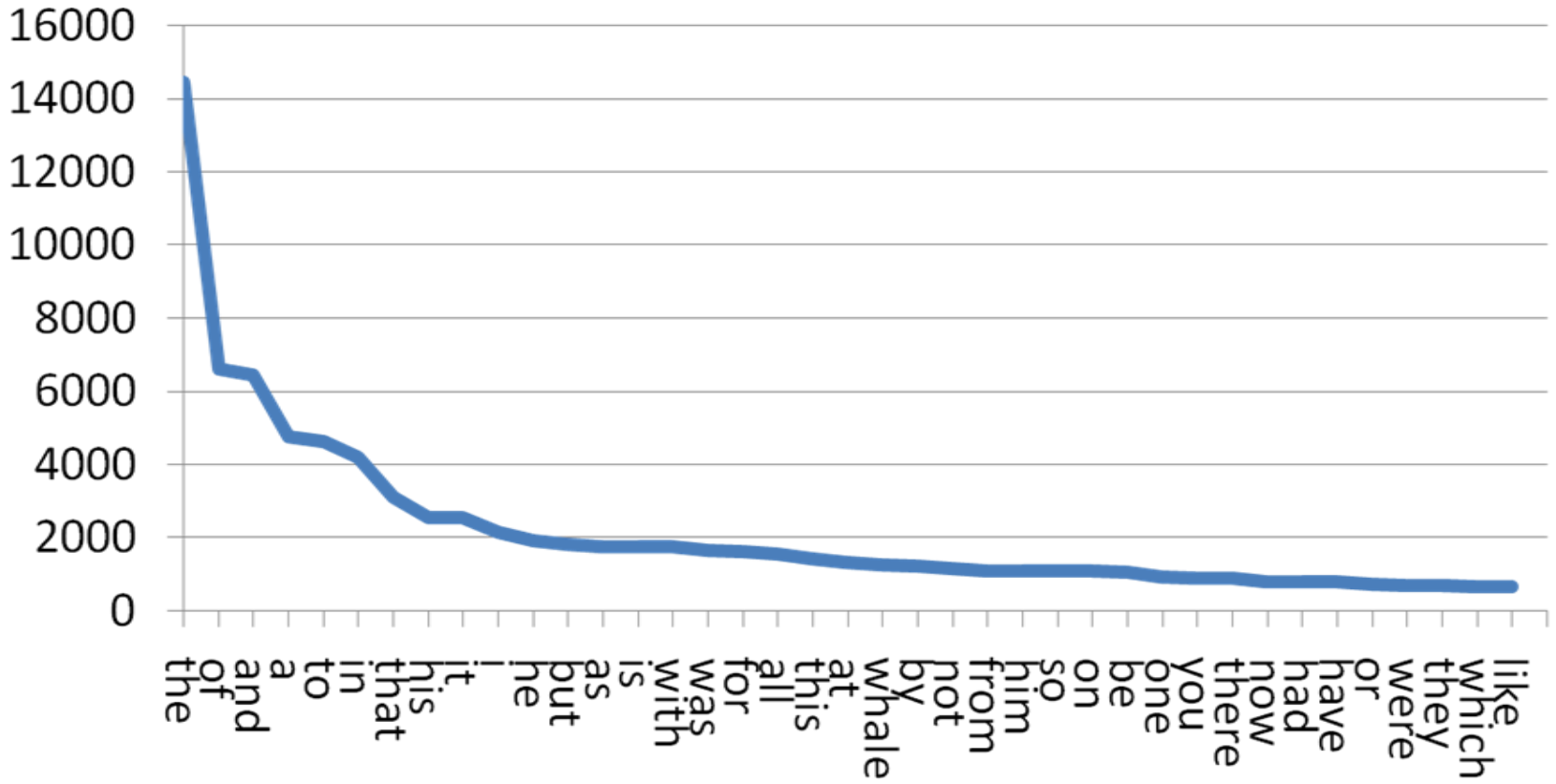
E	T	A	I	N	O	S	R	H	D	L	U	C
0.120	0.085	0.077	0.076	0.067	0.067	0.067	0.059	0.050	0.042	0.042	0.037	0.032

F	M	W	Y	P	B	G	V	K	Q	J	X	Z
0.024	0.024	0.022	0.022	0.020	0.017	0.017	0.012	0.007	0.005	0.004	0.004	0.002

Zipf's Law

Detour

- Term frequencies in *Moby Dick*:



Source: http://searchengineland.com/the-long-tail-of-search-12198

Zipf's Law

Detour

- The same is true for many other languages…
- **Zipf's own explanation:**
  - **Principle of least effort:**
    Do the job while minimizing total effort
  - **Cognitive effort of reading and writing should be small**
    ⇒ Pressure towards **unification of vocabulary** such that choosing and understanding words is easy (small vocabulary)
  - **Diversity of language has to be high**
    ⇒ Pressure towards **diversification of vocabulary** such that complex concepts can be expressed and distinguished
  - The **"economy of language"** leads to the balance observed and formalized by Zipf's law

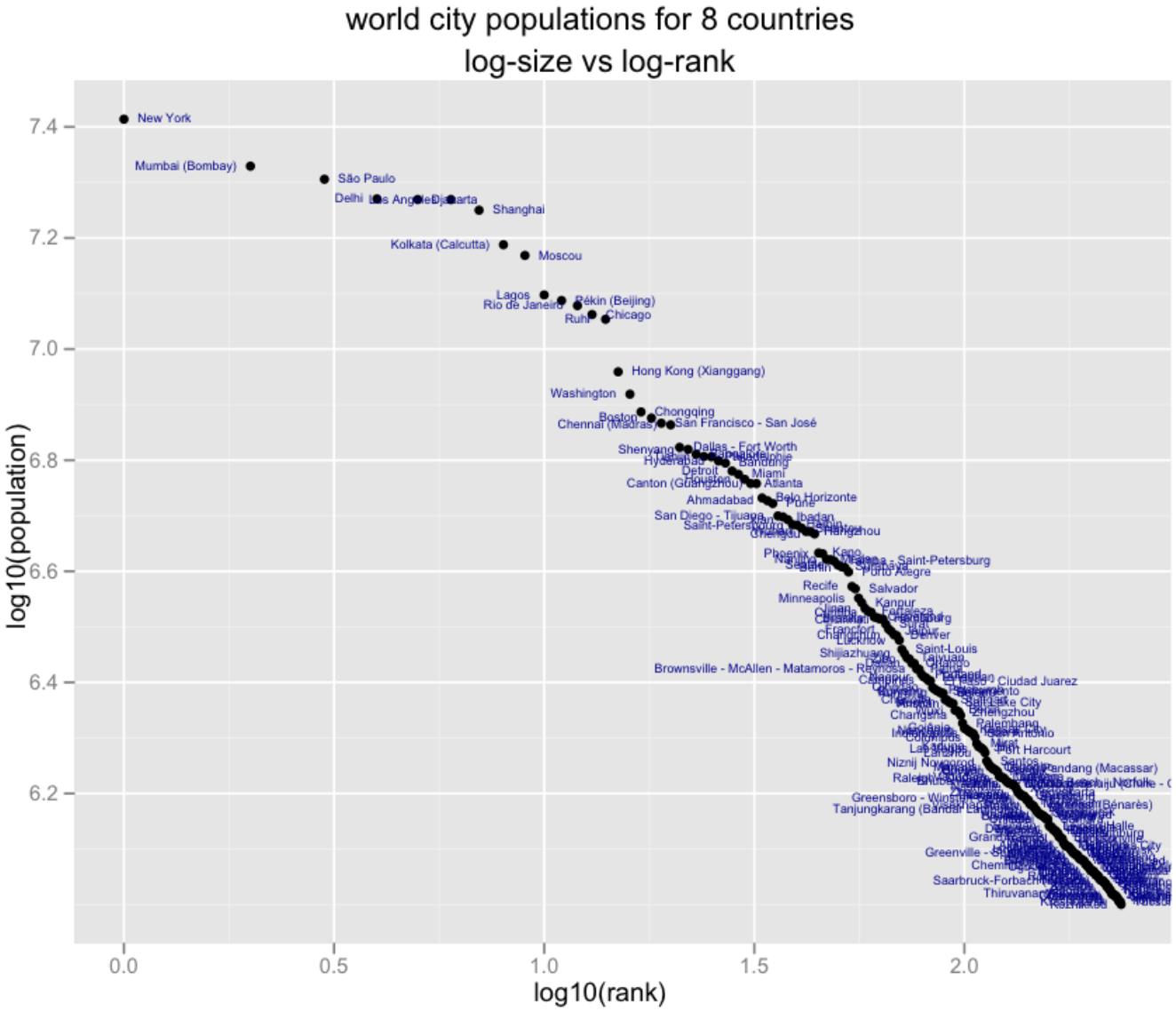
Zipf's Law

Zipf's law: The *i*-th most frequent term has frequency proportional to 1 / *i*

- Similar relationships hold in many different contexts:
  - Distribution of letter frequencies
  - Distribution of accesses per Web page
  - Distribution of links per Web page
  - Distribution of wealth
  - Distribution of population for US cities
  - …

Zipf's Law

Detour



Next Lecture

- Indexing
- Document normalization
  - Stemming
  - Stopwords
  - …
- Statistical properties of document collections