IST R&D PROJECT

SHARED-COST RTD PROJECT

THEME: INFORMATION ACCESS AND INTERFACES
COMMISSION OF THE EUROPEAN COMMUNITIES
DIRECTORATE GENERAL DG INFSO

PROJECT OFFICER: DR PAT MANSON

Resource Selection and Data Fusion for Multimedia
International Digital Libraries

Resource selection framework and methods

D3.1

February 25, 2002, UNIDO/WP3/Task 3.1/Version 2

Henrik Nottelmann, Norbert Fuhr

IST Project Number	IST-2000-26061	**Acronym**	MIND
Full title	Resource Selection and Data Fusion for Multimedia International Digital Libraries		
EU Project officer	Dr. Pat Manson		

Deliverable	**Number**	D3.1	**Name**	Resource selection framework and methods			
Task	**Number**	T3.1	**Name**	Resource selection framework			
Work Package	**Number**	WP3	**Name**	Resource selection			
Date of delivery	**Contractual**		2001/12/31	**Actual**		2002/02/25	
Code name	<codename>			**Version** 2	draft ☐	final ☑	
Nature	Prototype ☐ Report ☑ Specification ☐ Tool ☐ Other:						
Distribution Type	Public ☑ Restricted ☐ to: Partners, Commission, Reviewers						
Authors (Partner)	Henrik Nottelmann, Norbert Fuhr (UNIDO)						
Contact Person	Henrik Nottelmann						
	Email	nottelmann@ls6.cs.uni-dortmund.de	**Phone**	+49 231 755 5319	**Fax**	+49 231 755 2405	
Abstract (for dissemination)	This is the description of the MIND resource selection framework. It extends the decision-theoretic framework formerly developed by UNIDO by using a logistic function for modelling the relationship between score and probability of relevance, and by approximating the indexing weights (and, as a consequence, the document scores) by a normal distribution. This document includes a detailed description of the new framework and a first evalution.						
Keywords	Resource selection, decision-theoretic framework, logistic function, normal distribution, indexing weights distribution, score distribution.						

Contents

Chapter 1

Introduction

Networked information retrieval is a upcoming research area challenged by the rapid growth of the Internet and the widespread use of IR servers like e.g. WAIS or systems supporting the Z39.50 protocol (see e.g. [11]). Besides classical applications like bibliographies, more recently digital libraries have become accessible through the internet. The goal of the MIND project is to develop a system letting a user access all resources (digital libraries, DLs) available on the network, but in a way that gives him the impression of a single large IR system.

In the following, we assume a basic setting as follows: A user submits her query to a broker ("dispatcher" in the MIND system) which has access to a set of libraries to which it may send the query. In response, each library produces a ranked list of documents, and the broker may request any number of documents from this list; then the user is presented the merged output list. Resource selection is the task to determine the number of documents to be retrieved from each library (which should be zero for most the databases).

Some approaches – e.g., GlOSS [10, 9], CORI [2] – only regard retrieval quality as an optimisation criterion. In the MIND framework, costs are assigned to the retrieval of documents w. r. t. the library performance curve (in terms of recall and precision); in addition, the user attributes different costs to relevant and nonrelevant documents (wasted time) presented to her. Furthermore, time and monetary costs (for commercial libraries) are considered as well. This leads to a cost function consisting of a time, a monetary and a relevancy part. Then, the task is to find an optimum selection (for each library, the number of documents to retrieve from that library).

The MIND method is based on the decision-theoretic framework described in [7] (see chapter 2). Within the MIND project, we extended this framework:

1. The underlying retrieval model is changed (see chapter 3). The assumption of a linear relationship between the "score" $Pr(q \leftarrow d)$ and the probability $Pr(\text{rel}|q, d)$ that a document d is relevant w. r. t. a query q is revised (after some evaluation) and substituted by a logistic function. In our experiments, this logistic function more precisely modeled the relationship between score and probability of relevance.

2. Defining cost functions for time or monetary costs is quite easy. The difficult part is to estimate the number of relevant (irrelevant) documents in each single library result set. Here, we proposed two new methods (besides the recall-precision-method already described in [7]) for estimating this number (see section 4): simulating retrieval on a subset of the collection plus computing the document score distribution based on this result, and modeling indexing weights (and, thus, the document scores) by means of an exponential or a normal distribution.

We conducted a couple of experiments (see chapter 5), evaluating

1. if the indexing weights are distributed similar in the collection and the sample,
2. if the indexing weights are distributed according to an exponential or a normal distribution,
3. if the document scores are distributed similar in the collection and the sample,
4. if the document scores are distributed according to a normal distribution,
5. if a linear or a logistic function is more appropriate for modeling the relationship between score and probability of relevance,
6. if the number of relevant documents in a library can be estimated with the MIND framework,
7. if the number of relevant documents in a library's result set can be estimated with the MIND framework, and
8. if the overall retrieval quality of the new resource selection method improves quality.

In chapter 6, we transformed the cost function into an equivalent, but more user-friendly one (where the user has to state only three paramaters, each of them in $[0, 1]$ and uniquely defining the importance of time, money and retrieval quality).

Chapter 2

Decision-theoretic framework

The MIND framework starts from the Probability Ranking Principle [13], where it can be shown that optimum retrieval performance is achieved when documents are ranked according to decreasing probability of relevance. For resource selection, a decision-theoretic model which attributes different costs to the retrieval of relevant and nonrelevant documents in addition to other costs (e.g., computation or communication time, monetary costs for commercial libraries) is used [7]. The task is to minimise the overall costs for retrieval.

In detail, the MIND dispatcher has access to the libraries DL_1, DL_2, ... DL_m. Each of these DLs has its own query-specific cost function $C_i(s_i, q)$, where q is the query submitted by a user, and s_i is the number of documents to be retrieved from DL_i.

The task for the dispatcher is to compute an optimum solution, i.e. a vector $\vec{s} = (s_1, s_2, \ldots, s_m)^T$ with $|\vec{s}| = \sum_{i=1}^{n} s_i = n$ which minimises the overall costs:

$$M(n, q) := \min_{|\vec{s}|=n} \sum_{i=1}^{m} C_i(s_i, q).$$

The cost function $C_i : \mathcal{N} \times \mathcal{Q} \mapsto \mathcal{R}$ incorporates different cost sources:

- **Time:** The library has to be queried (mostly over the internet), the library has to perform its retrieval, and the results have to be sent back to the dispatcher. The time spent on these tasks is a cost factor C_i^T included in the resource selection framework. A simple approximation is an affin-linear function (with a constant part for submitting the query and a linear part for retrieval and returning the result documents):

$$C_i'^T(s_i, q) := C_{i,init}^T + s_i C_{i,doc}^T.$$

- **Money:** Many digital libraries integrated by MIND will be for free, but some DLs may charge. For a user, monetary costs may be one of the major issues which is covered by the cost function C_i^M. Once more, we can assume an affin-linear cost function (where mostly, the constant part equals zero):

$$C_i'^M(s_i, q) := C_{i,init}^M + s_i C_{i,doc}^M.$$

- **Effectiveness of the library:** A user is interested in getting many relevant documents. Thus, the effectiveness of the DLs is another important issue (probably the most important one). Let $r_i(s_i, q)$ denote the number of relevant documents in the result set when retrieving s_i documents from library DL_i with query q. Furthermore, we assign library-independent costs C^+ for viewing a relevant document and C^- for viewing a irrelevant document (with $C^+ < C^-$). This leads to the cost function

$$C_i^R(s_i, q) := r_i(s_i, q)C^+ + [s_i - r_i(s_i, q)]C^-.$$

These time and monetary cost factors must be modified slightly: for non-zero $C^T_{i,init}$ and $C^M_{i,init}$, we obtain $C'^T_i(0,q) > 0$ and $C'^M_i(s_i,q) > 0$, although the library does not contribute to the result (and, thus, should not be queried anyway). To correct this, we introduce the function

$$f : \mathcal{R} \mapsto \{0,1\}, f(s_i) := \begin{cases} 0 & , \quad s_i = 0 \\ 1 & , \quad \text{otherwise} \end{cases}$$

The costs für relevant and non-relevant documents don't have be corrected:

$$r_i(0,q) = 0 \ \wedge \ 0 - r_i(0,q) = 0 \implies C^R_i(0,q) = 0.$$

Another aim of the model is to allow for several user policies. E.g., one user might be interested in getting quick results and is willing to pay for them. Another user might be interested in getting many relevant documents at low monetary costs and is willing to wait for them.

The importance of effectiveness can be chosen by different values for C^+ and C^-. For time and money, we introduce two user-specific coefficients c^T and c^M, leading to

$$\begin{aligned} C^T_i(s_i,q) &:= c^T f(s_i) \cdot C'^T_i(s_i,q), \\ C^M_i(s_i,q) &:= c^M f(s_i) \cdot C'^M_i(s_i,q). \end{aligned}$$

The costs for retrieving s_i documents from DL_i with query q are:

$$C_i(s_i,q) := C^T_i(s_i,q) + C^M_i(s_i,q) + C^R_i(s_i,q).$$

As this cost function is required at resource selection time, i.e. before the libraries are queried, the system can only compute estimations. Thus, expected costs are used instead of the actual cost functions:

$$\begin{aligned} EC_i(s_i,q) &:= EC^T_i(s_i,q) + EC^M_i(s_i,q) + EC^R_i(s_i,q), \\ EC^T_i(s_i,q) &:= c^T f(s_i) \cdot \left[EC^T_{i,init} + s_i EC^T_{i,doc}\right], \\ EC^M_i(s_i,q) &:= c^M f(s_i) \cdot \left[EC^M_{i,init} + s_i EC^M_{i,doc}\right], \\ EC^R_i(s_i,q) &:= E[r_i(s_i,q)]C^+ + [s_i - E[r_i(s_i,q)]]C^-. \end{aligned}$$

Thus, the task is to minimise the expected overall costs:

$$EM(n,q) := \min_{|\vec{s}|=n} \sum_{i=1}^{m} EC_i(s_i,q).$$

2.1 Computing the expected costs

For computing the expected costs $EC_i(s_i,q)$, the system has to compute $EC^T_{i,init}$ and $EC^T_{i,doc}$ (for time), $EC^M_{i,init}$ and $EC^M_{i,doc}$ (for monetary costs) and $E[r(s_i,q)]$.

- **Time:** It is rather simple to estimate the time cost factors. During query-based sampling, test queries are sent to the library. The time from sending the query to receiving the result can be measured. Thus, $EC^T_{i,init}$ and $EC^T_{i,doc}$ can be approximated.
- **Money:** There seems to be no easy way to compute the query-independent monetary cost factors $EC^M_{i,init}$ and $EC^M_{i,doc}$. Thus, the person integrating the library should specify these values. Anyway, most libraries are for free: $EC^M_{i,init} = EC^M_{i,doc} = 0$.
- **Effectiveness of the library:** It is much more difficult to estimate the number of relevant documents in the result. Some methods are described in chapter 4.

2.2 Computing the optimum selection vector

With the cost function above, $EC_i(s_i, q)$ is monotonically increasing with s_i. Thus, the concept of dynamic programming can be applied, deriving the algorithm show in figure 2.1.

INPUT:

- number of documents to retrieve n,
- number of DLs m,
- $EC_i[k, q] := EC_i(k, q)$ for $1 \leq i \leq m$, $1 \leq k \leq n$

OUTPUT:

- minimum costs $EM[k] := EM(k, q)$ for $1 \leq k \leq n$,
- optimum selection vector $s[k][i]$ for retrieving $1 \leq k \leq n$ documents

ALGORITHM:

```
EM := EC_1
for k := 1 to n do
  s[k][1] := k
od
for i := 2 to m do
  (EM,s) := mincost(i,EM,EC_i,s)
od
return (EM,s);

function mincost(i,C1,C2,s);
S := s
for k := 0 to n do
  M[k] := C1[k];
  for j := 0 to k-1 do
    if M[k] > C1[j] + C2[k-j] then
      M[k] := C1[j] + C2[k-j];
      S[k] := s[j];
      S[k][i] := k-j:
    fi
  od
od
return (M,S);
```

Figure 2.1: Algorithm for computing optimum cost function

The outer loop iterates over all libraries. In the loop with variable `k` in `mincost`, the optimum costs and optimum selection vector are computed for retrieving `k` documents altogether. In the inner loop with variable `j`, all possibilities of retrieving `j` documents from $DL_1, \ldots, DL_{i-1}$ and `k-j` documents from DL_i. If the costs decrease, the selection vector `s[k]` has to be updated.

This algorithm needs $O(m \cdot n^2)$ time.

2.3 The tasks of the dispatcher and the proxies

The MIND architecture follows the standard structure with one mediator (called "dispatcher" in MIND) and wrappers ("proxies") for every library. The proxies extend the functionality of the non-co-operating libraries and give the dispatcher the required information not provided by the libraries. Thus, for the dispatcher this is the same scenario as if there are only co-operating libraries. The resulting architecture is depicted in figure 2.2.

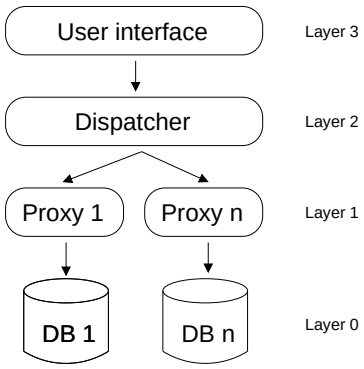


Figure 2.2: MIND architecture

Both dispatcher and proxies are responsible for resource selection. The proxies compute their expected costs $EC_i(s_i, q)$, i.e. a vector $(EC_i(0, q), EC_i(1, q), \ldots, EC_i(n, q))^T$. The dispatcher then uses these values to compute the optimum selection vector with one of the algorithms described in section 2.2.

Chapter 3

Computing relevance probabilities

As usual in modern IR, we view information retrieval as uncertain inference [17]. Thus, for a query q, the probability $Pr(\text{rel}|q,d)$ of a document being relevant can be formulated based on the probability of the uncertain implication $Pr(q \leftarrow d)$

$$Pr(\text{rel}|q,d) = f(Pr(q \leftarrow d))$$

with the query-specific function

$$f : [0,1] \mapsto [0,1].$$

For computing the implication probability $Pr(q \leftarrow d)$, the widely used linear retrieval function can be used [16, 18]:

$$Pr(q \leftarrow d) \quad = \quad \sum_{c_j \in q} \underbrace{Pr(q \leftarrow c_j)}_{\text{condition weight}} \cdot \underbrace{Pr(c_j \leftarrow d).}_{\text{indexing weight}}$$

Here, the query q consists of conditions c (for text retrieval, a condition simply is a term) with weight $Pr(q \leftarrow c_j)$. Furthermore, $Pr(c_j \leftarrow d)$ denotes the indexing weight (e.g., a normalised BM25 weight [14]).

Now, the expected number of relevant documents in a document set D can be estimated as follows:

$$\begin{aligned} E(\text{rel}|q,D) &= \sum_{d \in D} Pr(\text{rel}|q,d) \\ &= \sum_{d \in D} f(Pr(q \leftarrow d)). \end{aligned}$$

In practice, we do not have the probabilities $Pr(q \leftarrow d)$ (or $Pr(\text{rel}|q,d)$, respectively) for each document d. But, let us assume that $Pr(q \leftarrow d)$ is the score of the library search engine, and that we know the distribution of the scores. Then, we can compute the density p of the relevance probabilities $Pr(\text{rel}|q,d)$ (see figure 3.1).

First, we compute a point $a \in [0,1]$ which is the smallest score when retrieving s documents. This can be done via the formula:

$$s = |DL| \int_a^1 p(x)\ dx. \tag{3.1}$$

Then, we use this a to compute the expected number of relevant documents among these s retrieved documents:

$$r(s,q) = |DL| \int_a^1 p(x) \cdot x\ dx. \tag{3.2}$$

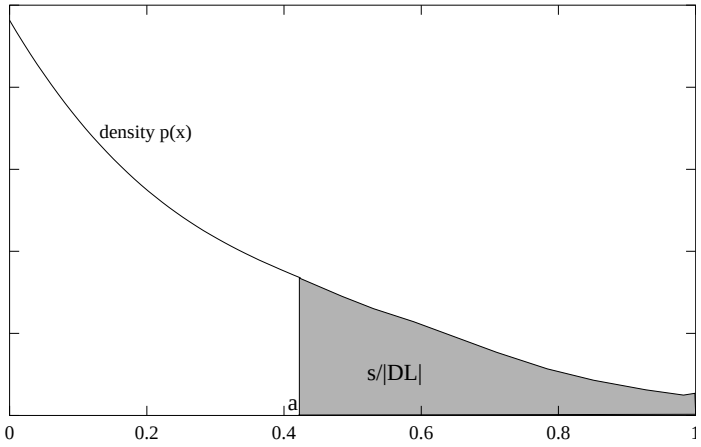


Figure 3.1: Density of the relevance probability distribution

The relationship between the probabilities $Pr(\text{rel}|q,d)$ and $Pr(\text{rel}|q \leftarrow d)$ is modelled by the function $f : [0,1] \mapsto [0,1]$. In this rest of this chapter, two different functions – namely a linear function and a logistic function – are described. Both functions are evaluated in section 5.4.

3.1 Linear function

The probability $Pr(\text{rel}|q,d)$ can be formulated based on the formula of total probabilities [17]:

$$\begin{aligned} Pr(\text{rel}|q,d) &= Pr(\text{rel}|q \leftarrow d) \cdot Pr(q \leftarrow d) + Pr(\text{rel}|\neg(q \leftarrow d)) \cdot Pr(\neg(q \leftarrow d)) \\ &\approx Pr(\text{rel}|q \leftarrow d) \cdot Pr(q \leftarrow d). \end{aligned}$$

Thus, the relationship between the probabilities $Pr(\text{rel}|q,d)$ and $Pr(q \leftarrow d)$ (the score) can be modelled as the linear function

$$f : [0,1] \mapsto [0,1], f(x) := Pr(\text{rel}|q \leftarrow d) \cdot x.$$

If no relevance data is available, only a constant can be assumed for estimating the query-specific conditional probability $Pr(\text{rel}|q \leftarrow d)$. With relevance data, we can apply Bayes' Rule

$$Pr(\text{rel}|q \leftarrow d) = \frac{Pr(q \leftarrow d|\text{rel}) \cdot Pr(\text{rel})}{Pr(q \leftarrow d|\text{rel}) \cdot Pr(\text{rel}) + Pr(q \leftarrow d|\neg\text{rel}) \cdot Pr(\neg\text{rel})},$$

where $Pr(q \leftarrow d|\text{rel})$ and $Pr(q \leftarrow d|\neg\text{rel})$ is the probability that a document implies the query given that it is relevant (non-relevant, respectively), and $Pr(\text{rel})$ is the probability that an arbitrary document is relevant.

Another solution would be to find a constant c which minimise the expected quadratic error

$$E\left[(Pr(\text{rel}|q,d) - c \cdot Pr(q \leftarrow d))^2\right]$$

and to estimate $Pr(\text{rel}|q \leftarrow d) \approx c$.

This can be done by means of least-square polynomials [15] and a learning sample (a set of documents d_i with scores $x_i := Pr(q \leftarrow d_i)$ and relevance judgements $y_i \in [0,1]$).

Now, the expected number of relevant documents in a document set D can be estimated as follows:

$$\begin{aligned} E(\text{rel}|q,D) &= \sum_{d\in D} Pr(\text{rel}|q,d) \\ &\approx Pr(\text{rel}|q \leftarrow d) \cdot \sum_{d\in D} Pr(q \leftarrow d) \\ &= Pr(\text{rel}|q \leftarrow d) \cdot \sum_{c_j \in q} Pr(q \leftarrow c_j) \cdot \left[\sum_{d\in D} Pr(c_j \leftarrow d)\right] \\ &= |D| Pr(\text{rel}|q \leftarrow d) \cdot \sum_{c_j \in q} Pr(q \leftarrow c_j) \cdot avg - weight(c_j,d) \end{aligned}$$

with the average indexing weight

$$avg - weight(c_j,d) := \frac{1}{|D|} \sum_{d\in D} Pr(c_j \leftarrow d).$$

Our experiments show that there is no linear relationship between the implication probability (score) $Pr(q \leftarrow d)$ and the probability of relevance $Pr(\text{rel}|q,d)$ (see section 5.4). Thus, other functions should be considered.

3.2 Logistic regression

In our experiments, mostly the top-ranked documents (with the highest score) are relevant. Thus, the function f (which should be a continuous function) should model this observation: Let $a \in [0,1]$ be a specific threshold. Then,

- $f(x) \approx 0$ for $x < a$,
- $f(x) \nearrow 1$ for $x \approx a$,
- $f(x) \approx 1$ for $x > a$.

This behaviour can be modeled by a logistic function [5, 6]:

$$f : [0,1] \mapsto [0,1],\ f(x) := \frac{\exp(b(x))}{1+\exp(b(x))} = 1 - \frac{1}{1+\exp(b(x))},$$

where $b(x)$ is a polynomial. For f, we choose

$$b : [0,1] \mapsto \mathcal{R},\ b(x) := b_0 + b_1 x.$$

The shape of the logistic function f is displayed in figure 3.2.

To compute the parameters (b_0, b_1), a learning sample (documents d_i with scores $x_i := Pr(q \leftarrow d_i)$ and relevance judgement $y_i \in \{0,1\}$) are required. Furthermore, we extend $f(x)$ to $f(x, b_0, b_1)$.

As optimisation criterion, maximum likelihood is used [8]. Thus, the goal is to maximise the likelihood function

$$\begin{aligned} L(b_0,b_1) &:= \prod_{y_i=1} f(x_i,b_0,b_1) \prod_{y_i=0} (1 - f(x_i,b_0,b_1)) \\ &= \prod_i f(x_i,b_0,b_1)^{y_i} (1 - f(x_i,b_0,b_1))^{1-y_i}. \end{aligned}$$

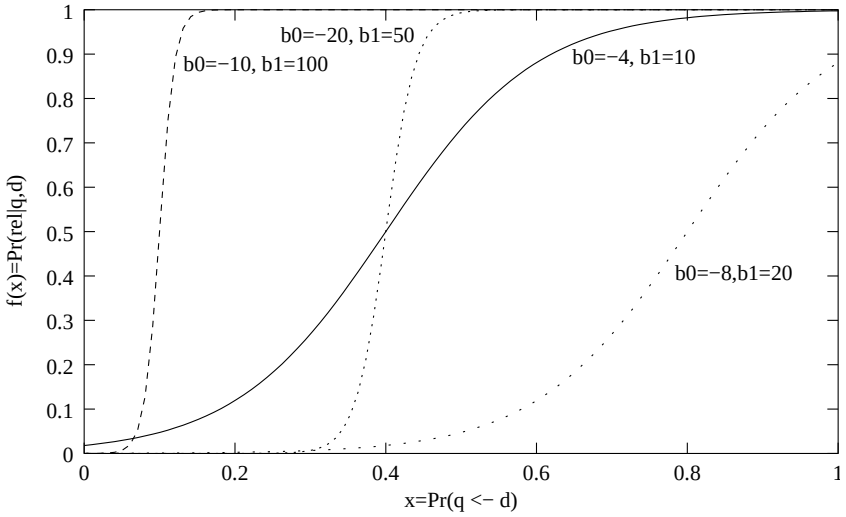


Figure 3.2: Logistic function with different parameters b_0, b_1

In the following, the log-likelihood function $l(b_0, b_1)$ is regarded:

$$l(b_0, b_1) := logL(b_0, b_1) = \sum_i y_i \log f(x_i, b_0, b_1)^{y_i} + (1 - y_i) \log(1 - f(x_i, b_0, b_1)).$$

A necessary criterion for a maximum is

$$\frac{\partial l}{\partial b_j} = \sum_i (y_i - f(x_i, b_0, b_1)) x_{i_j} =: g_j(b) = 0.$$

In general, this equation system cannot be solved in closed form. Thus, we apply the iterative Newton-Raphson method where we need the Hesse matrix

$$H_{jk}(b_0, b_1) := \frac{\partial^2 l}{\partial b_j \partial b_k} = -\sum_i f(x_i, b_0, b_1)(1 - f(x_i, b_0, b_1)) x_{i_j} x_{i_k}.$$

Starting from a vector $\vec{b}^{(0)}$ (a reasonable good approximations), we apply the iteration formula

$$\vec{b}^{(n+1)} = \vec{b}^{(n)} - H^{-1}(\vec{b}^{(n)}) g(\vec{b}^{(n)})$$

until we reach the stopping criterion

$$\max_j \left| \frac{b_j^{(n+1)} - b_j^{(n)}}{b_j^{(n+1)}} \right| < \varepsilon$$

with a predefined value $\epsilon > 0$.

With this method, the parameters b_0, b_1 of the logistic function f can be computed (based on a learning sample). This will done a preparation step (directly after QBS).

The experiments proofed that this is a much better approximation (see section 5.4).

Chapter 4

Estimating the number of relevant retrieved documents

This chapter describes three methods for estimating the number $r_i(s_i, q)$ of relevant documents in the result set of library DL_i for query q (where s_i documents are to be retrieved).

All these methods are based on the same retrieval model described in the previous chapter.

4.1 Method 1: Recall-precision function

This method is described in detail in [7]. We assume a linear relationship function f as described in section 3.1. Then, we can compute

$$R_i := E(\text{rel}|q, DL_i) = |DL_i| Pr(\text{rel}|q \leftarrow d) \cdot \sum_{c_j \in q} Pr(q \leftarrow c_j) \cdot avg - weight_i(c_j, d)$$

with the average indexing weight

$$avg - weight_i(c_j, d) := \frac{1}{|DL_i|} \sum_{d \in D} Pr(c_j \leftarrow d).$$

Our goal is to compute the expected precision $EP_i(s_i, q)$ when retrieving s_i documents from DL_i. Then, $E[r_i(s_i, q)] = s_i \cdot EP_i(s_i, q)$.

For estimating the expected precision, we use an affin-linear recall-precision function (see figure 4.1)

$$P_i : [0, 1] \mapsto [0, 1], \; P(R) := P_i^0 \cdot (1 - R).$$

Although this is a query-specific function, without additional relevance judgements we can only consider a global approximation. With relevance judgements, least-square polynomials [15] can be used for estimating the parameter P_i^0.

Given the recall-precision-function P_i and the expected number R_i of relevant documents in the library, and with $r = E[r_i(s_i, q)]$, we get

$$\begin{aligned} EP_i(s_i, q) = \frac{r}{s} &= P_i(\frac{r}{R_i}) = P_i^0(1 - \frac{r}{R_i}) \\ r &= \frac{P_i^0 R_i s}{R_i + P_i^0 s}. \end{aligned}$$

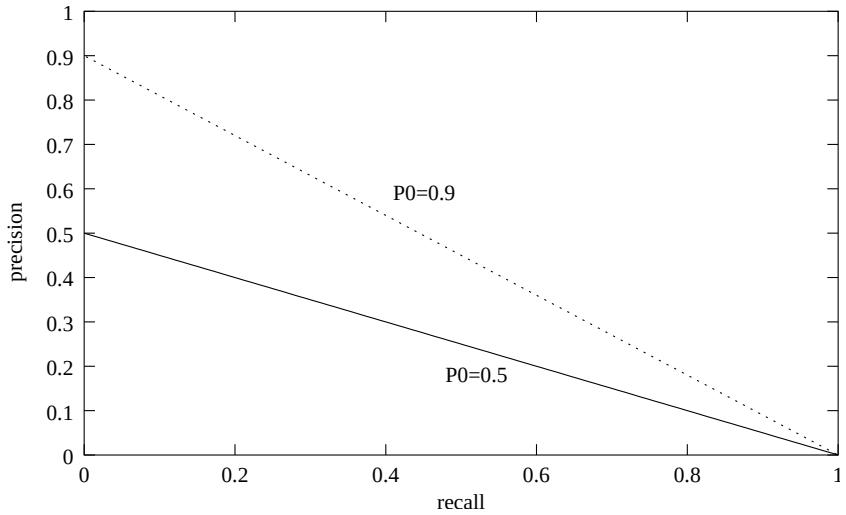


Figure 4.1: Recall-precision-functions

4.2 Method 2: Simulated retrieval on sample

With query-based sampling [1], "random" subsequent queries are submitted to the library in order to obtain an unbiased sample of that library. With reasonable low costs (i.e., number of queries), it is possible to retrieve an sufficiently unbiased sample of e.g. 300 documents. For other resource selection algorithms like CORI [2], statistical resource descriptions like average term frequencies avg_tf_t and inverse document frequencies df_t are extracted. Experiments showed that these resource descriptions are sufficiently similar to the "real" resource description for CORI.

For this method, the complete retrieved sample is used. The documents are indexed (with stemming and stop word elimination), and all terms t with the corresponding indexing weights $Pr(t \leftarrow d)$ (e.g., BM 25) is stored in a database or IR system. Like the original QBS technique, this will be done in a preparation step for MIND (which only has to be done once, but it is a good idea to repeat this after some time to incorporate changes of the underlying library).

Given a query q, the resource selection module simulates retrieval on this small sample (300 documents), obtaining a score (probability $Pr(q \leftarrow d)$) for each sample document. These scores are used to compute the score distribution density. Then, the expected number of relevant retrieved documents can be computed be means of the function f and the formulae 3.1 and 3.2.

The advantages of this method are:

- The method is very easy to understand and to implement.
- The method uses information from the query-based sampling process which is necessary anyway.
- This method is suitable for all media types.

An disadvantage is that we have to simulate retrieval on this sample *before* querying the actual library. In section 5.7, the efficiency of this method is evaluated.

4.3 Method 3: Modelling term indexing weights

This method tries to model the distribution of the indexing weights. Then, the indexing weight distribution is used to model the score distribution.

The indexing weight of a term t is the probability $Pr(t \leftarrow d)$. Let p_t be the density of the indexing weights for term t in all documents in the library. Then, the aim is to find an approximation of this density.

We evaluated two different, well known distributions, the exponential distribution (subsection 4.3.1) and the normal distribution (subsection 4.3.2), the results are shown in section 5.2.

For a query with two (independent) terms t_1 and t_2 with weight densities p_1, p_2, the indexing weights for these terms in the library are considered as random variables $X_j := Pr(t_j \leftarrow d)$. Then, together with the abbreviations $a_j := Pr(q \leftarrow t_j)$, we obtain the resulting random variable X (the scores)

$$X := a_1 \cdot X_1 + a_2 \cdot X_2 = Pr(q \leftarrow d).$$

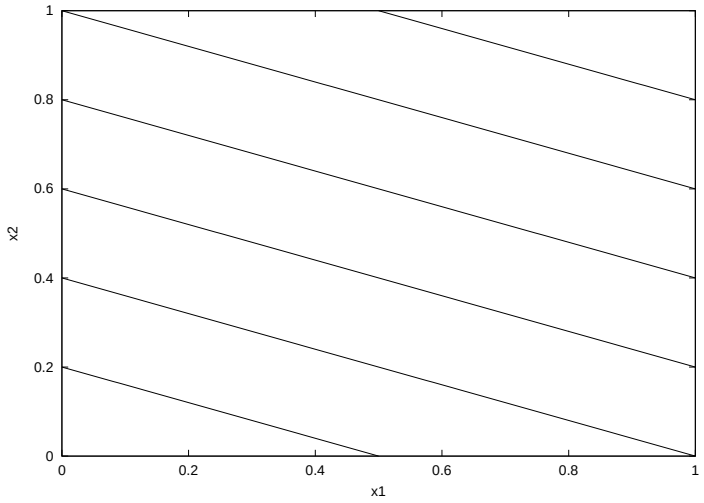


Figure 4.2: Indexing weights for terms t_1 and t_2, lines with constant $Pr(q \leftarrow d)$

For the density p of the score random variable X, $p(x)$ is the integral over the line

$$a_1 \cdot x_1 + a_2 \cdot x_2 = x$$

(depicted in figure 4.2). Thus, $p(x)$ can be computed as

$$p(x) = \int_{-\infty}^{\infty} \frac{1}{a_2} p_1(x_1) p_2(\frac{1}{a_2}(x - a_1 \cdot x_1))\ dx_1.$$

4.3.1 Exponential distribution

The exponential distribution (see figure 4.3) has the only parameter $\lambda = E[x]^{-1}$ (inverse of the expectation):

$$p_t(\mathit{weight} = x) = \frac{\lambda}{1 - \exp(-\lambda)} \exp(-\lambda x) 1_{[0,1]}(x)$$

The distribution of X, a linear combination of exponentially distributed random variables, is no exponential distribution. The resulting, rather complex formula is given in appendix A. For more than two terms, the density should be computed in a numerical way.

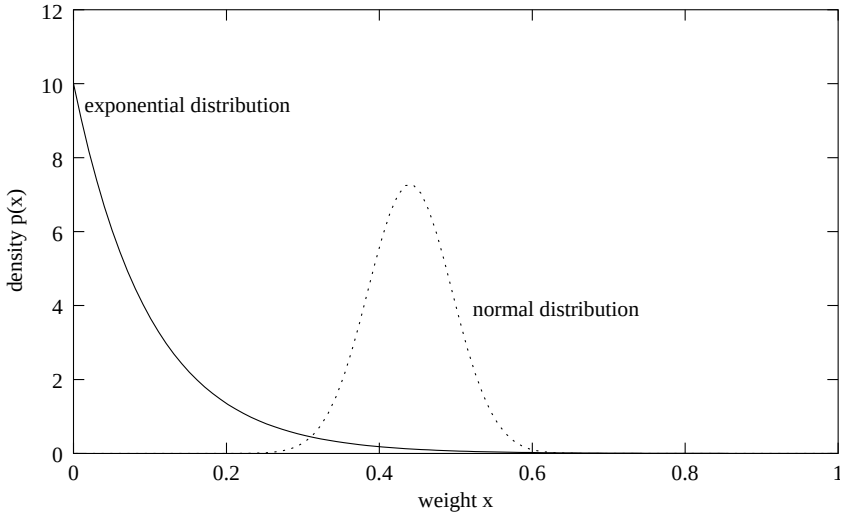


Figure 4.3: Exponential distribution with $\lambda = 10$, normal distribution with $\mu = 0.44$, $\sigma^2 = 0.003$

4.3.2 Normal distribution

The normal distribution (see figure 4.3) has two parameters $\mu = E[x]$ (expectation) and $\sigma^2 = V[x] = E[(x - E[x])^2]$ (variance):

$$p_t(weight = x) := \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

Here, the distribution density p of the scores X is again a normal distribution with the parameters

$$\begin{aligned} \mu = E[X] &= a_1 \cdot E[X_1] + a_2 \cdot E[X_1], \\ \sigma^2 = V[X] &= (a_1)^2 \cdot V[X_1] + (a_2)^2 \cdot V[X_2]. \end{aligned}$$

For a query with l terms, we obtain a normal distribution with the parameters

$$\mu = E[X] = \sum_{i=1}^{l} a_i \cdot E[X_i], \tag{4.1}$$

$$\sigma^2 = V[X] = \sum_{i=1}^{l} (a_i)^2 \cdot V[X_1]. \tag{4.2}$$

This, the computation of the parameters μ and σ^2 is easy and efficient computations.

The assumption of normally distributed indexing weights leads to normally distributed scores which contrast Manmatha's work in [12]. According to this paper, there is a normal distribution for the scores of the relevant documents and an exponential distribution for the irrelevant documents. But in practice, it is often impossible to compute the corresponding distribution parameters (about 60 relevant documents are required to compute the parameters, but in most cases a digital library does not contain so many relevant documents at all). Thus, we restricted our work to the easier case of a normal distribution. Our experiments in section 5.2 support this assumption.

Chapter 5

Evaluation

This chapter describes the evaluation of the MIND resource selection framework, its basic hypotheses (normal distribution, logistic relationship function f), the three methods for estimating the number of relevant retrieved documents and the efficiency of method 2.

5.1 Test bed

We performed evaluation with the CMU 100 collection test bed. To reduce the evaluation time, we only used the AP 1988-1990 collection and split it into 24 collections, according to the CMU 100 collection partition [4].

We removed all TREC stop words, and applied the Porter stemmer. As indexing weights, we used a BM 25 variant (for original BM 25, see [14])

$$\begin{aligned} tfw(t,d) &:= \frac{tf(t,d)}{tf(t,d) + 0.5 + 1.5 \cdot \frac{dl(d)}{avgdl}} \\ idf(t) &:= \frac{\log \frac{|DL|}{df(t)}}{\log |DL_{collection}|} \\ P(t \leftarrow d) &:= tfw(t,d) \cdot idf(t). \end{aligned}$$

Here, $tf(t,d)$ denotes the number of times term t appeared in document d (term frequency), $dl(d)$ the number of tokens in d (document length), $avgdl$ the average document length in the collection, $df(t)$ the number of documents containing term t, $|DL|$ the number of documents in document set $|DL|$ (collection or sample), and $|DL_{collection}|$ the number of documents in the complete collection.

Different to other authors, we normalised the idf component s.th. the indexing weight in the closed interval $[0,1]$ and, thus, can be regarded as the probability $Pr(t \leftarrow d)$).

Furthermore, we used the 300 document QBS samples specified by [3].

As test queries, we used the `description` fields TREC topics 51-100 (with stemming and stop word removal). One of the topic texts is depicted in figure 5.1. Relevance judgements for these queries are given within the TREC collection. Documents with no judgement are treated as irrelevant.

As query weights $Pr(q \leftarrow t)$, we used the normalised tf

$$P(q \leftarrow t) := \frac{tf(t,q)}{ql(q)}.$$

Here, $tf(t,q)$ denotes the number of times term t appears in the query q, and $ql(q)$ the number of tokens in q (query length). Thus, $\sum_{t \in q} P(q \leftarrow t) = 1$.

```
<top>

<head> Tipster Topic Description

<num> Number:  051

<dom> Domain:  International Economics

<title> Topic:  Airbus Subsidies

<desc> Description:
Document will discuss government assistance to Airbus Industrie, or mention a
trade dispute between Airbus and a U.S. aircraft producer over the issue of
subsidies.

<smry> Summary:
Document will discuss government assistance to Airbus Industrie, or mention a
trade dispute between Airbus and a U.S. aircraft producer over the issue of
subsidies.

<narr> Narrative:
A relevant document will cite or discuss assistance to Airbus Industrie by the
French, German, British or Spanish government(s), or will discuss a trade
dispute between Airbus or the European governments and a U.S. aircraft
producer, most likely Boeing Co. or McDonnell Douglas Corp., or the U.S.
government, over federal subsidies to Airbus.

<con> Concept(s):
1. Airbus Industrie
2. European aircraft consortium, Messerschmitt-Boelkow-Blohm GmbH, British
   Aerospace PLC, Aerospatiale, Construcciones Aeronauticas S.A.
3. federal subsidies, government assistance, aid, loan, financing
4. trade dispute, trade controversy, trade tension
5. General Agreement on Tariffs and Trade (GATT) aircraft code
6. Trade Policy Review Group (TPRG)
7. complaint, objection
8. retaliation, anti-dumping duty petition, countervailing duty petition,
   sanctions

<fac> Factor(s):

<def> Definition(s):

</top>
```

Figure 5.1: TREC topic 51

5.2 Indexing weights

In method 2 – simulated retrieval on sample –, the indexing weights in the sample are used as approximations of the indexing weights in the collection for estimating the document score distribution. Thus, first of all it should be verified that the indexing weight distributions are similar in both the collection and the sample.

Method 3 – modeling term indexing weights – aims to fit the actual indexing weights of the terms by a well-known distribution (see section 4.3). Candidates are the exponential and the normal distribution distribution; both distributions are evaluated.

The results can be found in figures 5.2 (exponential distribution) and 5.3 (normal distribution). The first plot in each diagram shows the actual indexing weights in the complete `ap88_1` collection (10.586 documents), the second plot the corresponding (exponential or normal) fit. The two other plots show the actual indexing weights in the 300 document sample and the corresponding fit, respectively. The parameters of the fit (λ for the exponential fit, μ and σ^2 for the normal fit) are computed from the collection and sample, respectively. As there is a huge number of documents which do not contain a term (and, thus, have a zero indexing weight), there is a big peak at zero which is omitted in the plots for better readability.

Although there are slight differences, the indexing weight distributions seem to be rather similar in both the collection and the sample. Furthermore, it became clear that the exponential distribution is not appropriate for modelling the BM 25 indexing weights. In contrast, the normal distribution seems to be appropriate as an approximation. The results are very much the same for the other collections and terms.

5.3 Document scores

In this framework, a document score is the implication probability $Pr(q \leftarrow d)$.

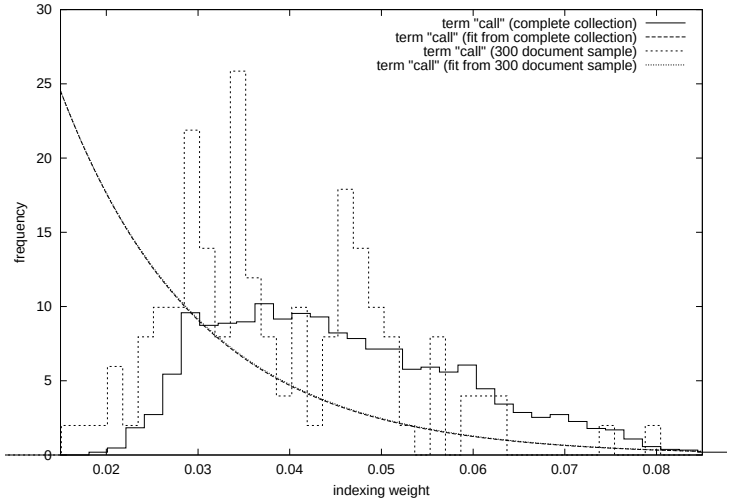
Both, method 2 and method 3, are evaluated:

- In method 2, retrieval is simulated on the sample, and the resulting distribution is used as an approximation of the actual document score distribution in the collection.
- In method 3, the parameters (μ,σ^2) of the document score distribution are computed based on the normal distribution of the indexing weights in the sample (see subsection 4.3.2, formulae 4.1 and 4.2):

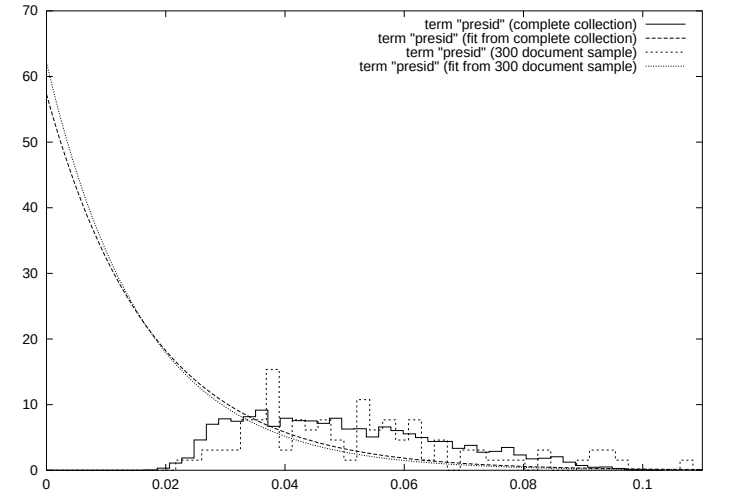
$$\begin{aligned} \mu = E[X] &= \sum_{i=1}^{l} a_i \cdot E[X_i], \\ \sigma^2 = V[X] &= \sum_{i=1}^{l} (a_i)^2 \cdot V[X_1]. \end{aligned}$$

The results can be found in figure 5.4. The first plot in each diagram shows the actual document scores in the complete `ap88_1` collection (10.586 documents), the second plot the resulting normal distribution based on the indexing weight distribution parameters of the collection. The two other plots show the actual document scores in the 300 document sample and the resulting normal distribution based on the indexing weight distribution parameters of the sample, respectively. As there is a huge number of documents with zero score, there is a big peak at zero which is omitted in the plots for better readability.

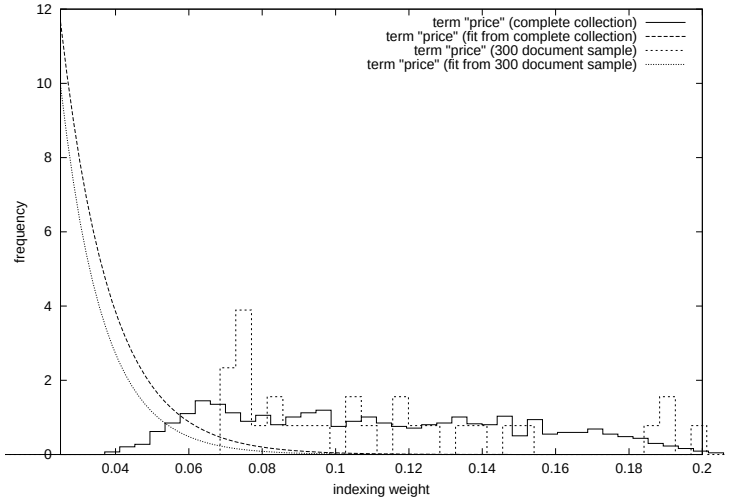
The distributions in the collection and the sample are quite similar. Furthermore, a normal distribution seems to be a good approximation for the document scores, although there are some differences at the high (and most interesting scores). Once more, the results are very much the same for the other collections and queries.



Term "call": $\lambda_{coll} = 66.2238$, $\lambda_{sample} = 65.6283$

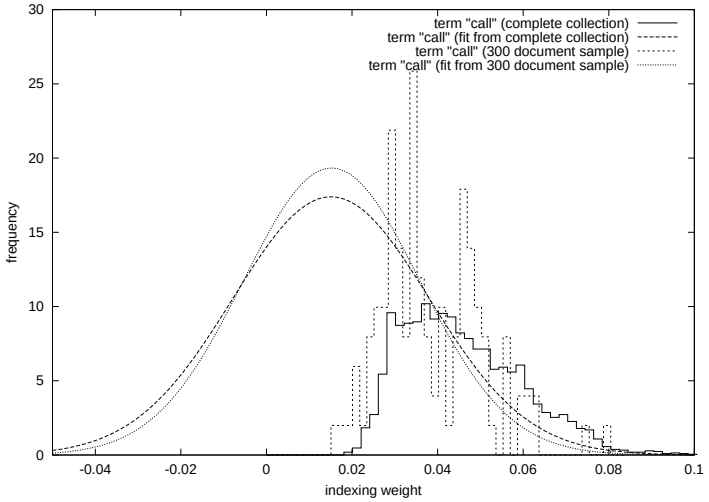


Term "presid": $\lambda_{coll} = 57.2959$, $\lambda_{sample} = 62.1471$

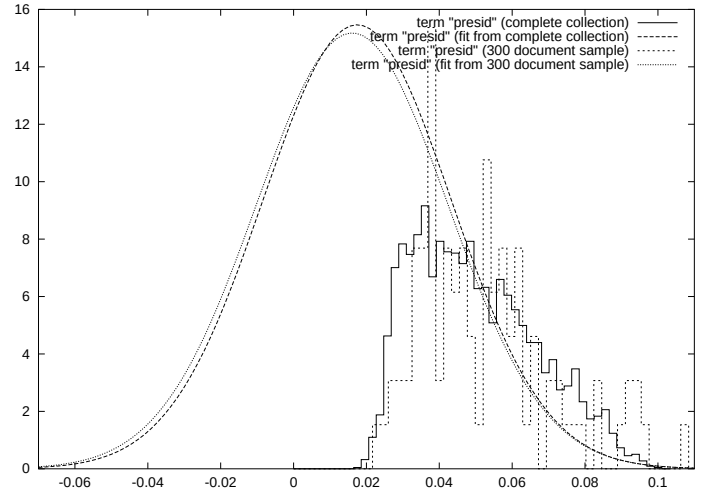


Term "price": $\lambda_{coll} = 73.8588$, $\lambda_{sample} = 86.5250$

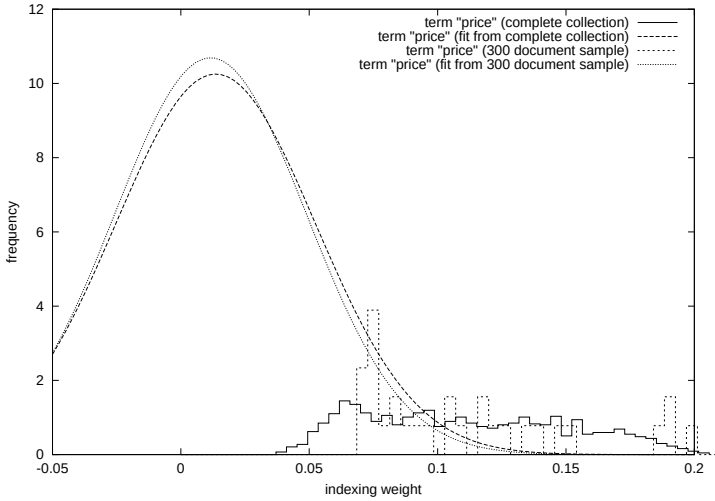
Figure 5.2: Indexing weights – exponential distribution



Term "call": $\mu_{coll} = 0.015100$,$\sigma^2_{coll} = 0.000525$, $\mu^2_{sample} = 0.015237$,$\sigma^2_{sample} = 0.000425$

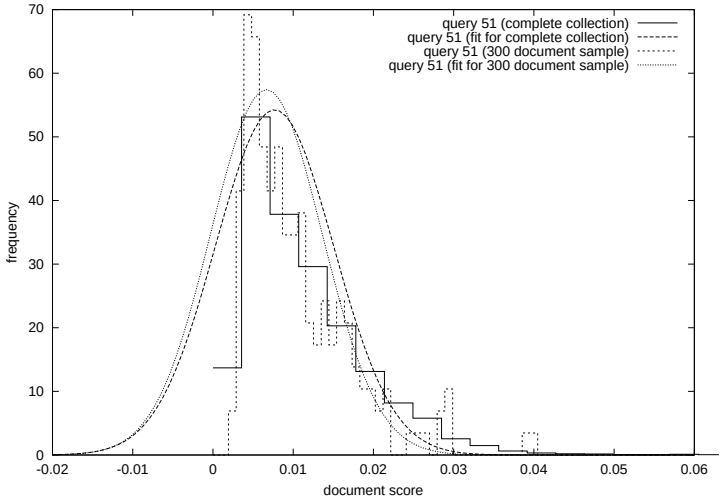


Term "presid": $\mu_{coll} = 0.017453$,$\sigma^2_{coll} = 0.000665$, $\mu^2_{sample} = 0.016090$,$\sigma^2_{sample} = 0.000690$

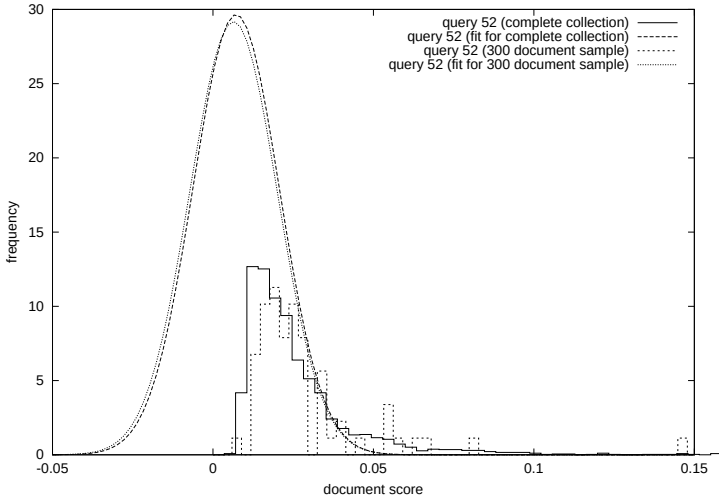


Term "price": $\mu_{coll} = 0.013539$,$\sigma^2_{coll} = 0.001514$, $\mu^2_{sample} = 0.0115573$,$\sigma^2_{sample} = 0.001392$

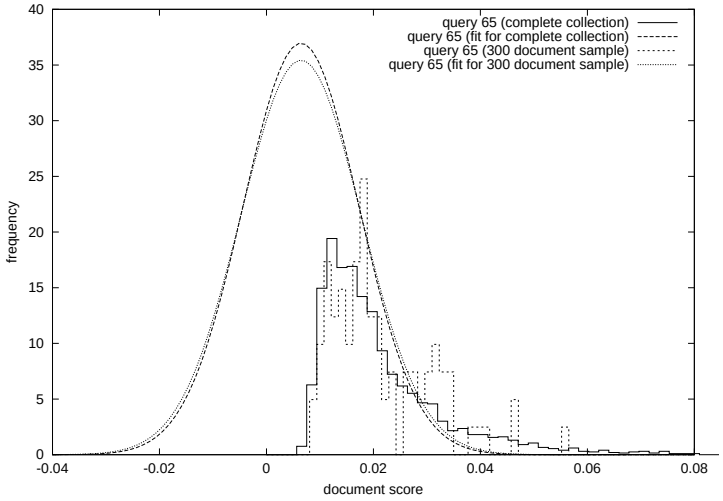
Figure 5.3: Indexing weights – normal distribution



Query 51: $\mu_{coll} = 0.007668$,$\sigma^2_{coll} = 5.405478\text{e-}05$, $\mu^2_{sample} = 0.006660$,$\sigma^2_{sample} = 4.826139\text{e-}05$



Query 52: $\mu_{coll} = 0.007217$,$\sigma^2_{coll} = 0.000180$, $\mu^2_{sample} = 0.006400$,$\sigma^2_{sample} = 0.000186$



Query 65: $\mu_{coll} = 0.006456$,$\sigma^2_{coll} = 0.000116$, $\mu^2_{sample} = 0.006462$,$\sigma^2_{sample} = 0.000126$

Figure 5.4: Document scores – normal distribution

5.4 Computing relevance probabilities

In chapter 3, we proposed two different functions for modeling the relationship between the score $Pr(q \leftarrow d)$ and the probability of relevance $Pr(\text{rel}|q,d)$:

Linear function:

$$f : [0,1] \mapsto [0,1],\ f(x) := Pr(\text{rel}|q \leftarrow d) \cdot x.$$

Logistic function:

$$f : [0,1] \mapsto [0,1],\ f(x) := \frac{\exp(b_0 + b_1 x)}{1 + \exp(b_0 + b_1 x)}.$$

In this section, we evaluate both functions. Plots are shown for the complete `ap88_1` collection and its 300 document sample, both for parameters which are optimal for that query and global parameters (computed for all queries). The results (actual relevance judgements and estimated probabilities of relevance) are depicted in figures 5.5, 5.6, 5.7 and 5.8.

One can see that the quality of the linear function is very poor, whereas the logistic function is an appropriate approximation for the underlying relationship (e.g., for query 51, the estimated values nearly exactly match the relevance judgements) if the relevant documents are the top-ranked ones.

Furthermore, it becomes clear that for the sample, the quality also is poor for the logistic functions, The reason is that the highest score in the sample is much below the highest score in the collection, and thus has a probability of nearly zero. We will see that the quality of method 2 is bad.

5.5 Number of relevant documents in the DL

Method 1 relies on estimating the number of relevant documents in the library. Thus, we tested the quality of this number against the actual number of relevant documents in the library (derived from the relevance judgements). For method 1 (recall-precision function), we used a linear relationship between score and probability of relevance, for method 2 and 3 we used a logistic function. For method 3, we first normalised the scores s.th. the highest normalised score is 1, as this improves the quality. For method 2, this had no significant impact, thus we used unnormalised scores. In all cases, we computed global parameters.

We tested estimations from the sample and the collection. Some results are shown in figures 5.1 and 5.2.

collection	query	actual	method 1	method 2	method 3
ap88_1	51	5	5	1	9
ap88_1	52	19	5	21	6
ap88_1	64	10	6	2	8

Table 5.1: Expected number of relevant documents in DL (sample)

collection	query	actual	method 1	method 2	method 3
ap88_1	51	5	6	7	9
ap88_1	52	19	5	38	6
ap88_1	64	10	7	5	8

Table 5.2: Expected number of relevant documents in DL (collection)

These tables shows a poor quality, even for the optimum estimation. This result gives a first hint about a poor quality of method 1, the only method for which the expected number of relevant documents in the underlying library is necessary.

5.6 Number of relevant documents in result set

All methods aim to estimate the number $r_i(s_i, q)$ of relevant documents in the result set of DL_i when retrieving s_i documents. Thus, we tested all three methods (optimum test with estimations derived from the collection and real application tests with estimations derived from the sample only).

For method 1, the recall-precision-function (or, more precisely, the starting parameter P_i^0) is computed from a learning sample with least-square polynomials [15]. For our experiments, we used all relevance judgements for all queries included in the TREC collection.

Some results are shown in figure 5.9 (sample) and 5.10 (collection). The first line denotes the actual relevance judgements (baseline).

5.7 Efficiency

One potential performance bottleneck of method 2 is the simulated retrieval on the library sample (300 documents).

Thus, we measured the real time of this retrieval run on collection `ap88_1` with query `51` (13 terms) and a short version (3 manually created term) of it. Evaluation was done with a small perl script, and all data was stored in a mysql database, running on two distinct Linux PC machines.

We measured the time for 1,000 retrieval runs, and repeated that test several times. The average time for one retrieval run (on the 300 documents) is about 50 ms. This number did not vary much with the number of terms involved.

For comparison, we also measured time for retrieval on the complete library ($\approx$ 10,000 documents). Here, one retrieval run takes about 2.5 s.

5.8 Overall retrieval quality

In reality, were are not particularly interested in estimating the expected number of relevant documents in one library's result set, but we want to maximise the number of relevant documents in the fused result set.

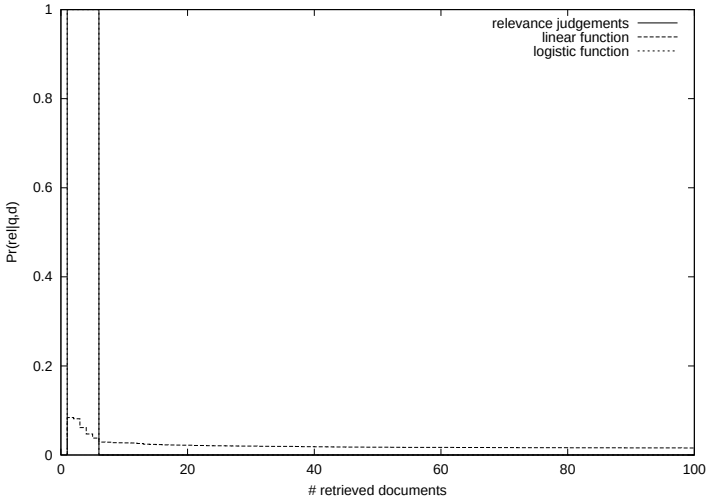
We evaluated all three methods (where the estimations are derived from the samples only, and global parameters for the function f are used). For the cost function, we simply set $EC_i^T \equiv 0$ and $EC_i^M \equiv 0$, thus we considered only retrieval quality.

Some results can be seen in figure 5.11. Here, the very first line shows the baseline, the result of resource selection using the actual number of relevant documents (derived from the relevance judgements). The very last line shows the result of a retrieval run over the union of all 24 AP collections.

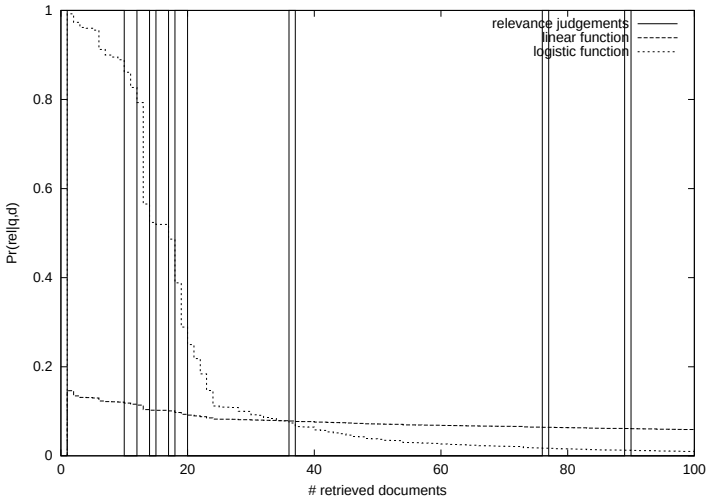
One can see that method 1 (the one proposed in the former publication [7]) and method 3 (the new method described in this document) perform quite similar, with slight improvements through the new method 3 for some queries. Method 2 performs worse.

On the other hand, the quality of method 1 and 3 is in most cases far away from the optimum baseline.

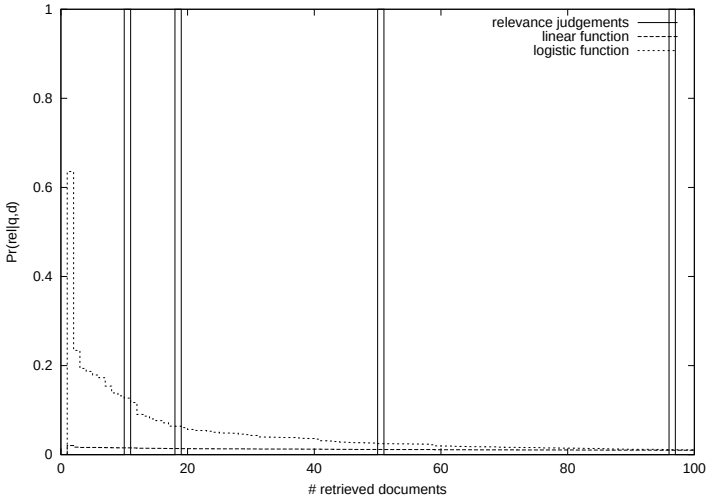
As another baseline, we also conducted experiments with estimations based on the complete collection. Although not shown in the plots, retrieval quality does not significantly increase for method 3. Thus, the loss by estimating from a sample only is quite small. This is not surprising as the statistical moments of the normal distributions are very similar in the collection and the sample.



Query 51

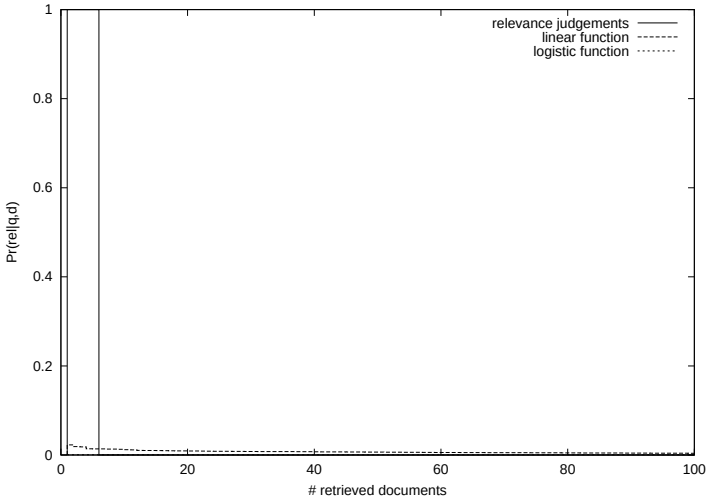


Query 52

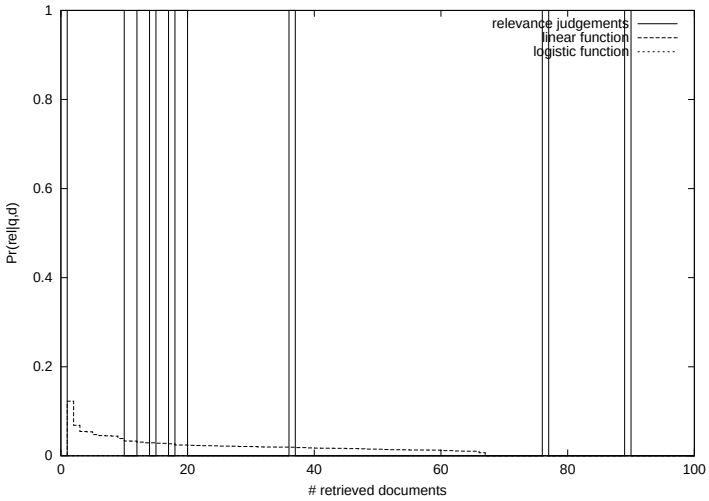


Query 64

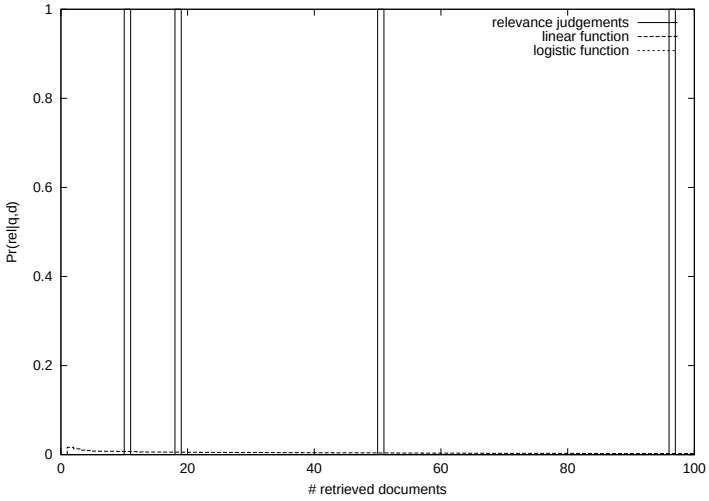
Figure 5.5: Relevance probabilities (query-specific parameters, collection)



Query 51

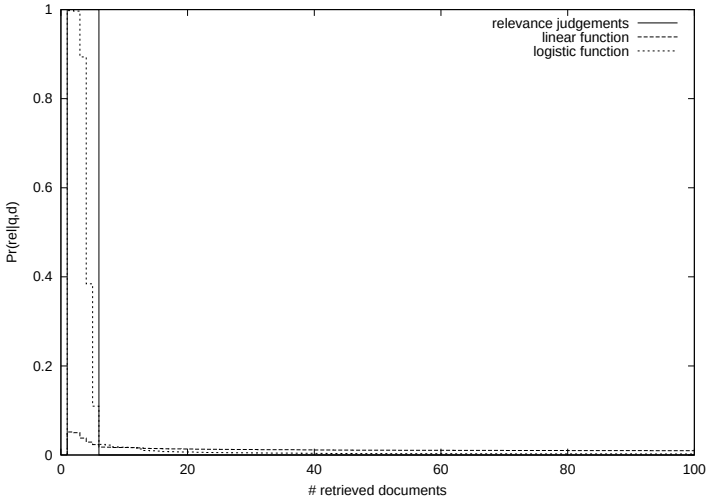


Query 52

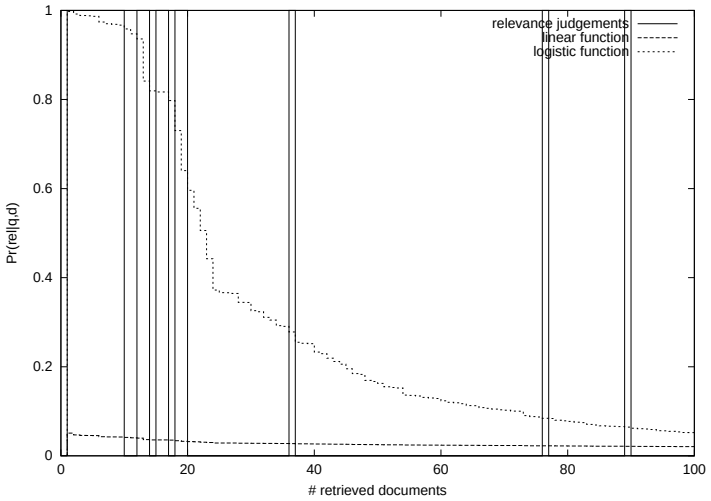


Query 64

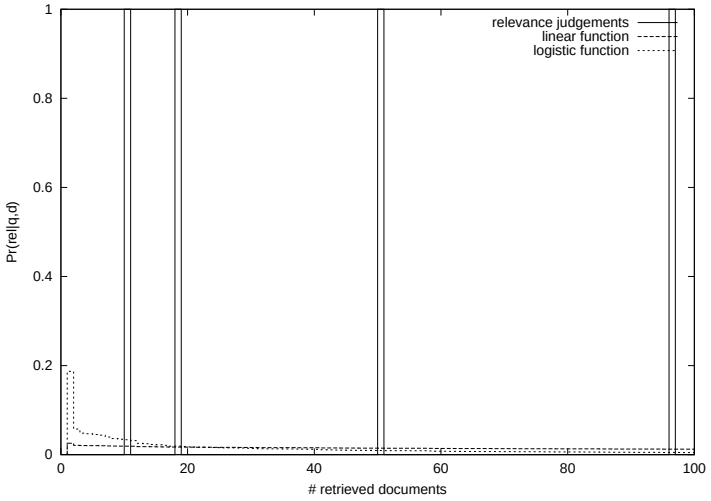
Figure 5.6: Relevance probabilities (query-specific parameters, sample)



Query 51

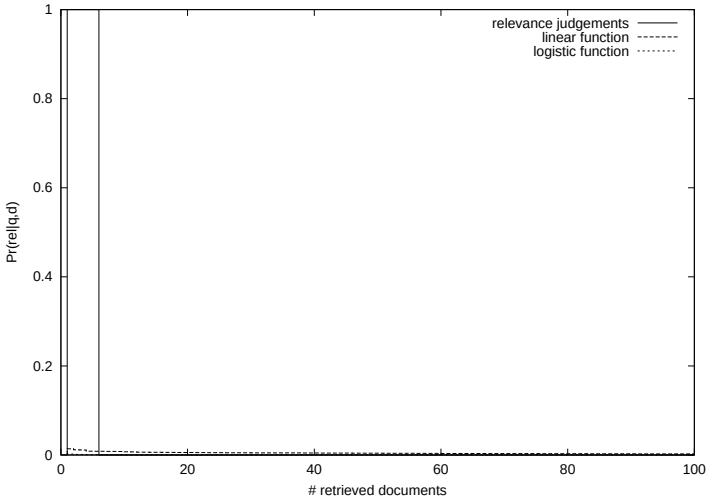


Query 52

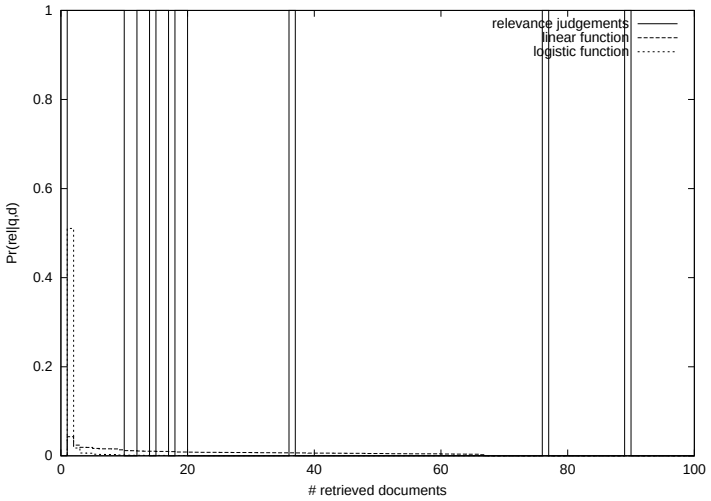


Query 64

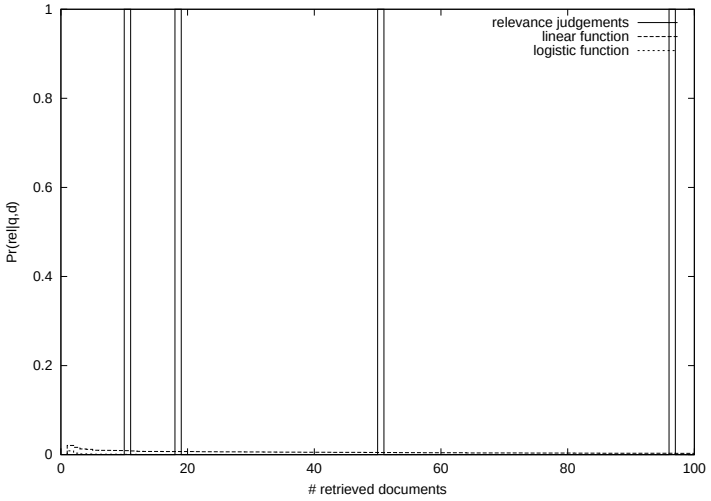
Figure 5.7: Relevance probabilities (global parameters, collection)



Query 51

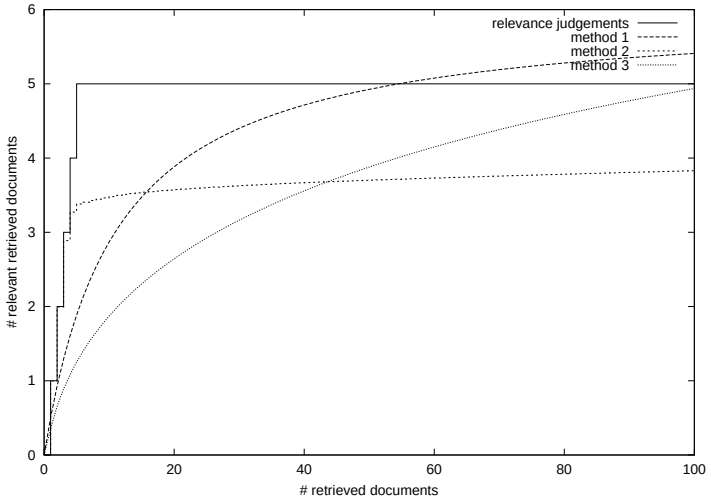


Query 52

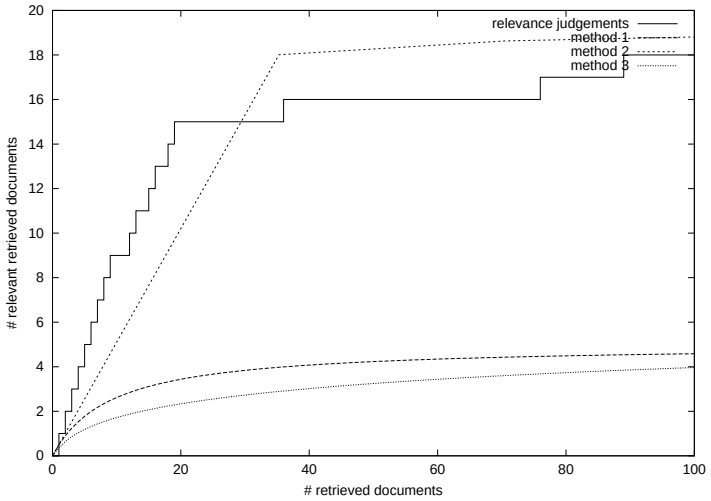


Query 64

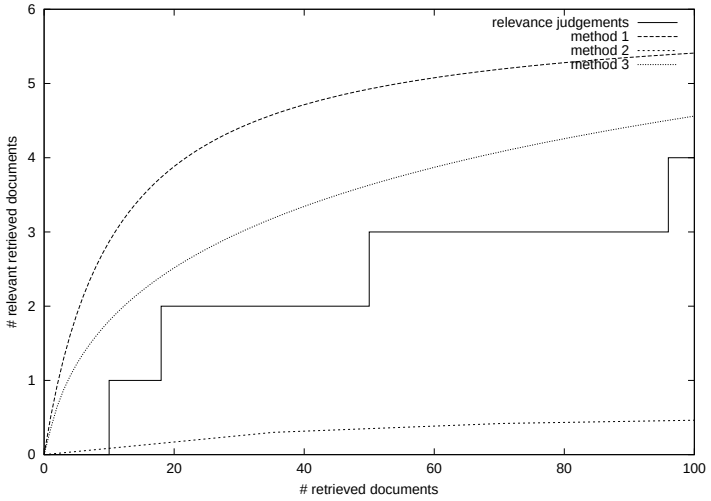
Figure 5.8: Relevance probabilities (global parameters, sample)



Query 51

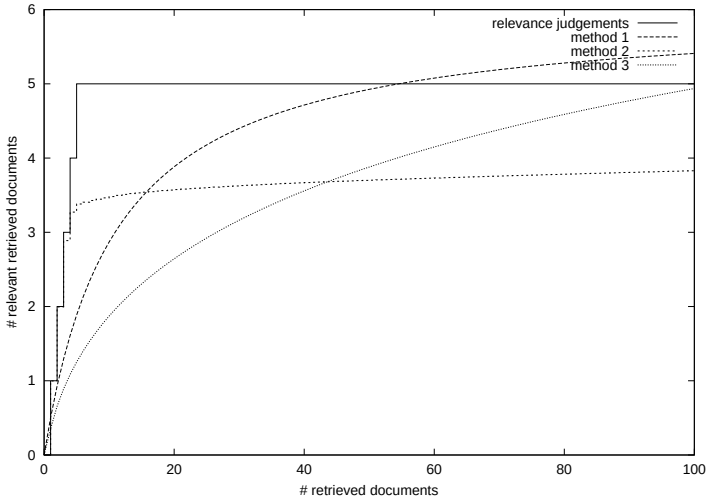


Query 52

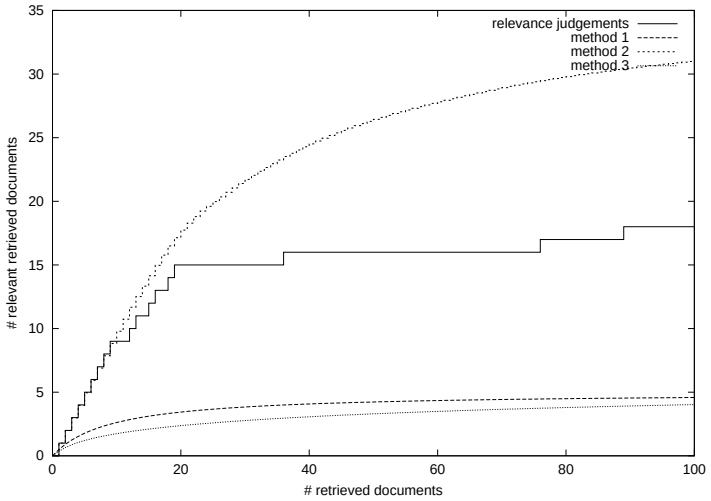


Query 64

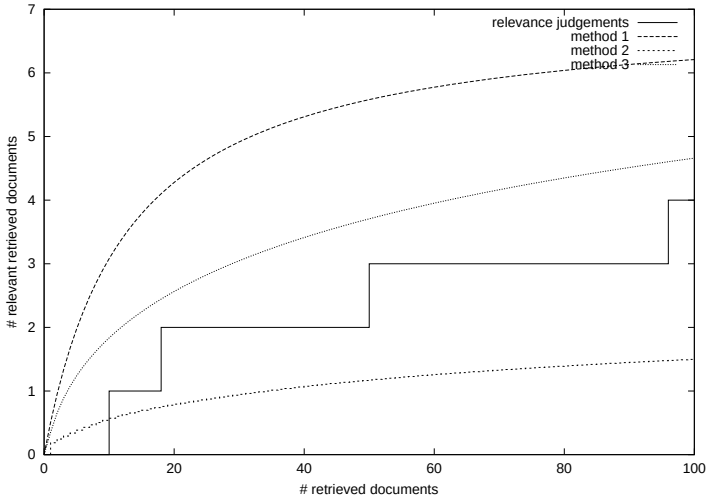
Figure 5.9: Expected number of relevant retrieved documents (sample)



Query 51

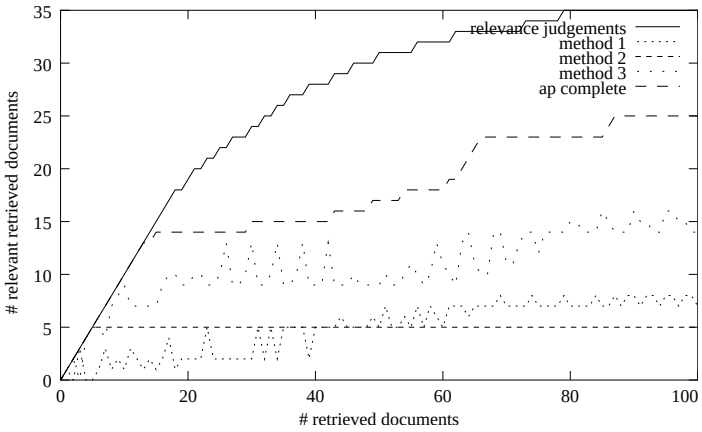


Query 52

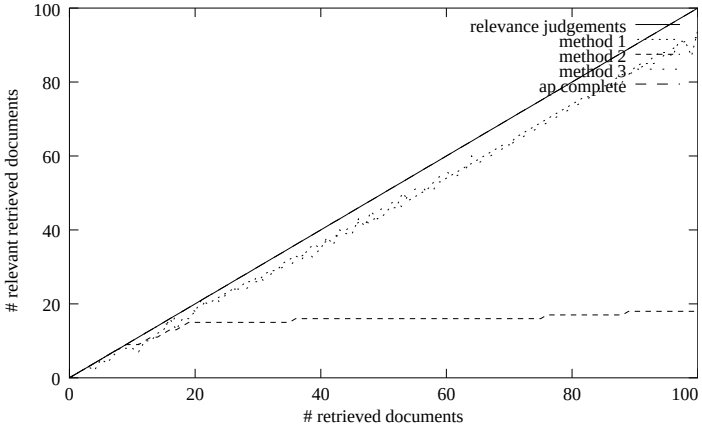


Query 64

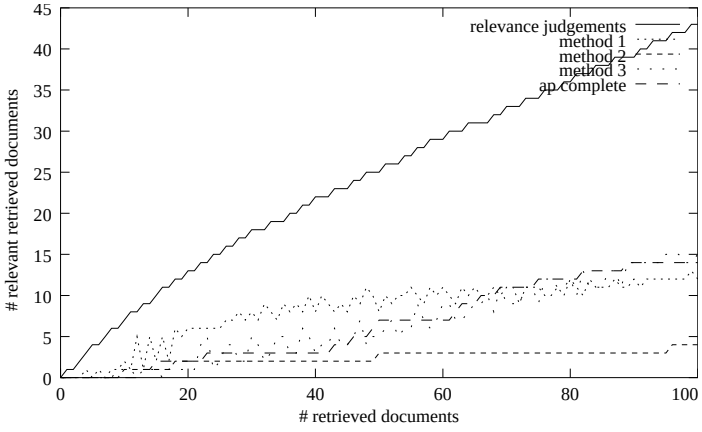
Figure 5.10: Expected number of relevant retrieved documents (collection)



Query 51



Query 52



Query 64

Figure 5.11: Overall retrieval quality

Chapter 6

User-friendly cost parameters

The expected costs for retrieving s_i documents from DL_i (corresponding to query q) are

$$\begin{aligned} EC_i(s_i,q) &:= EC_i^T(s_i,q) + EC_i^M(s_i,q) + EC_i^R(s_i,q), \\ EC_i^T(s_i,q) &:= c^T f(s_i) \cdot \left[EC_{i,init}^T + s_i EC_{i,doc}^T\right], \\ EC_i^M(s_i,q) &:= c^M f(s_i) \cdot \left[EC_{i,init}^M + s_i EC_{i,doc}^M\right], \\ EC_i^R(s_i,q) &:= E[r_i(s_i,q)]C^+ + [s_i - E[r_i(s_i,q)]]C^-. \end{aligned}$$

Here, EC^T denotes the time required for retrieval (computation and communication time), EC^M monetary costs (for free DLs, it is zero), and EC^R describes the relevancy of that DL. Of course, other functions besides the affin-linear one can be considered as well for the time and the monetary part. Furthermore, we have the user-defined mixing parameter c^T, c^M, C^+ and C^- allowing for different user policies (e.g., "cheap results", "good results", "fast results").

The goal is to minimise the expected overall costs

$$EM(n,q) := \min_{|\vec{s}|=n} \sum_{i=1}^{m} EC_i(s_i,q).$$

For a user, it is difficult to specify the mixing parameters correctly. It would be an advantage if all mixing parameters could be in $[0,1]$ and could have a well-defined and well-understandable semantics. This means that we want to minimise

$$c^T T_i(s_i,q) + c^M M_i(s_i,q) - c^R R_i(s_i,q) \text{ with } c^T, c^M, c^R \in [0,1],$$

where $T_i(s_i,q) \in [0,1]$ denotes the time part, $M_i(s_i,q) \in [0,1]$ the monetary part and $R_i(s_i,q) \in [0,1]$ the relevancy part (where many relevant documents are a benefit in contract to costs for time and money).

Definition 6.1 *Normalisation parameters are defined as follows:*

$$\begin{aligned} T_{norm} &:= m^{-1} \max_i \{EC_{i,init}^T + nEC_{i,doc}^T\}^{-1}, \\ M_{norm} &:= m^{-1} \max_i \{EC_{i,init}^M + nEC_{i,doc}^M\}^{-1}, \\ R_{norm} &:= m^{-1} n^{-1}. \end{aligned}$$

Definition 6.2 *The new time, monetary and relevancy parts are defined as:*

$$\begin{aligned} T_i(s_i,q) &:= T_{norm} \cdot f(s_i) \cdot \left[EC_{i,init}^T + s_i EC_{i,doc}^T\right], \\ M_i(s_i,q) &:= M_{norm} \cdot f(s_i) \cdot \left[EC_{i,init}^M + s_i EC_{i,doc}^M\right], \\ R_i(s_i,q) &:= R_{norm} \cdot E[r_i(s_i,q)]. \end{aligned}$$

Definition 6.3 *The overall time, monetary and relevancy parts are defined as:*

$$T(\vec{s},q) := \sum_{i=1}^{m} T_i(s_i,q),\ M(\vec{s},q) := \sum_{i=1}^{m} M_i(s_i,q),\ R(\vec{s},q) := \sum_{i=1}^{m} R_i(s_i,q).$$

Theorem 6.4

$$T(\vec{s},q) \in [0,1], M(\vec{s},q) \in [0,1], R(\vec{s},q) \in [0,1] \text{ for } 0 \leq s_i \leq n.$$

Proof

1. Time part $T(\vec{s},q)$: (analogous proof for monetary part $M(\vec{s},q)$)

$$\begin{aligned} T(\vec{s},q) &:= \sum_{i=1}^{m} T_i(s_i,q) = T_{norm} \cdot \sum_{i=1}^{m} f(s_i) \cdot \left[EC^T_{i,init} + s_i EC^T_{i,doc}\right] \\ &= m^{-1} \max_i \{EC^T_{i,init} + nEC^T_{i,doc}\}^{-1} \cdot \sum_{i=1}^{m} f(s_i) \cdot \left[EC^T_{i,init} + s_i EC^T_{i,doc}\right] \\ &\leq m^{-1} \max_i \{EC^T_{i,init} + nEC^T_{i,doc}\}^{-1} \cdot \sum_{i=1}^{m} \left[EC^T_{i,init} + nEC^T_{i,doc}\right] \leq 1. \end{aligned}$$

2. Relevancy part $R(\vec{s},q)$:

$$R(\vec{s},q) := \sum_{i=1}^{m} R_i(s_i,q) = \sum_{i=1}^{m} m^{-1}n^{-1}E[r_i(s_i,q)] \leq m^{-1}n^{-1}\sum_{i=1}^{m} s_i \leq m^{-1}n^{-1}\sum_{i=1}^{m} n = 1.$$

$\diamond$

Definition 6.5 *For constants $c^T \in [0,1]$, $c^M \in [0,1]$ and $c^R \in [0,1]$, we define*

$$U(\vec{s},q) := c^T T(\vec{s},q) + c^M M(\vec{s},q) - c^R R(\vec{s},q).$$

With $U(\vec{s},q)$, we get a cost function split into three different parts, each of them is in $[0,1]$ (i.e., they are normalised), and with new user-specific cost parameters $c^T \in [0,1]$, $c^M \in [0,1]$ and $c^R \in [0,1]$.

Now, we can define an equivalent but more human-readable optimisation problem:

Theorem 6.6 *The two optimisation problems are equivalent:*

$$EM(n,q) := \min_{|\vec{s}|=n} \sum_{i=1}^{m} EC_i(s_i,q) \tag{6.1}$$

$$U(m,q) := \min_{|\vec{s}|=n} c^T T(\vec{s},q) + c^M M(\vec{s},q) - c^R R(\vec{s},q) \tag{6.2}$$

Proof

1. "(6.1) ⇒ (6.2)": Let $\vec{s}$ be an optimum solution for eqn 6.1. With

$$c := c^T T_{norm}^{-1} + c^M M_{norm}^{-1} + n(C^- - C^+),$$

we obtain:

$$\begin{aligned}
\sum_{i=1}^{m} EC_i(s_i,q) &:= \sum_{i=1}^{m} EC_i^T(s_i,q) + EC_i^M(s_i,q) + EC_i^R(s_i,q) \\
&= \sum_{i=1}^{m} c^T f(s_i) \cdot \left[EC_{i,init}^T + s_i EC_{i,doc}^T\right] + \\
&\quad c^M f(s_i) \cdot \left[EC_{i,init}^M + s_i EC_{i,doc}^M\right] + \\
&\quad E[r_i(s_i,q)]C^+ + [s_i - E[r_i(s_i,q)]]C^- \\
&= c\sum_{i=1}^{m} \left[\underbrace{\frac{c^T T_{norm}^{-1}}{c}}_{:=c'^T \leq 1} \cdot T_i(s_i,q)\right] + \left[\underbrace{\frac{c^M M_{norm}^{-1}}{c}}_{:=c'^M \leq 1} \cdot M_i(s_i,q)\right] + \\
&\quad \left[s_i C^- - \underbrace{\frac{n(C^- - C^+)}{c}}_{:=c'^R \leq 1} \cdot R_i(s_i,q)\right] \\
&= c\sum_{i=1}^{m} s_i C^- + c\left[\sum_{i=1}^{m} c'^T T_i(s_i,q) + \sum_{i=1}^{m} c'^M M_i(s_i,q) - \sum_{i=1}^{m} c'^R R_i(s_i,q)\right] \\
&= cnC^- + c\left[T(\vec{s},q) + c^M M(\vec{s},q) - c^R R(\vec{s},q)\right] \\
&= cnC^- + c \cdot U(\vec{s},q)
\end{aligned}$$

   Thus, as $\vec{s}$ is optimal for eqn 6.1, it is also optimal for eqn 6.2.

2. "(6.2) ⇒ (6.1)": vice versa.

◇

For the new optimisation problem

$$U(m,q) := \min_{|\vec{s}|=n} c^T T(\vec{s},q) + c^M M(\vec{s},q) - c^R R(\vec{s},q),$$

the user only has to adjust the mixing parameters $c^T \in [0,1]$, $c^M \in [0,1]$ and $c^R \in [0,1]$. It is quite easy to construct a very user-friendly interface for setting up the mixing parameters (e.g., with the slider solution depicted in figure 6.1).

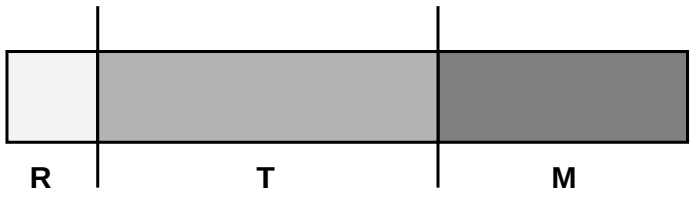


Figure 6.1: Slider solution to adjust c^R, c^T and c^M

Chapter 7

Conclusion and outlook

In this document, we introduced the MIND resource selection framework. This method bases on the decision-theoretic framework described in [7].

We applied two major extensions:

1. Instead of a linear relationship between score $Pr(q \leftarrow d)$ and probability of relevance $Pr(\text{rel}|q, d)$ with factor $Pr(\text{rel}|q \leftarrow d)$, we used (see section 3.2) a logistic function

$$\frac{\exp(b_0 + b_1 x)}{1 + \exp(b_0 + b_1 x)}.$$

2. For estimating the number of relevant documents in a library's result set, we computed document score distributions (from a simulated retrieval on a sample – section 4.2 – or derived from normally distributed indexing weights – section 4.3) instead of a recall-precision-function.

Our evaluation justified both extensions. The logistic function much better describes the relationsip between score and probability of relevance. Approximating the indexing weights and document scores by means of a normal distribution (method 3) improves the overall retrieval quality for some queries, for other queries it performs quite similar to the old method. Method 2 – simulated retrieval on sample – performs worse than method 3 and, thus, should not be persued any longer.

Further on, the overall retrieval quality of both the old and the new method still is noticeably below the optimum solution (actual relevance judgements). Thus, the estimation of the number of relevant document has to be improved further.

Thus, we will continue studying the relationship between score $Pr(q \leftarrow d)$ and probability of relevance $Pr(\text{rel}|q, d)$. Although our experiments are quite promising (particularly if the top-ranked documents are the relevant ones), the quality decreases when global parameters (b_0, b_1) for the logistic function are taken in contrast to query-specific ones. Thus, we will investigate query-specific normalisation factors for the parameters which could be computed by means of relevance feedback.

We will further try to improve the quality of the estimation process. One big problem is that the normal distribution is a good approximation for the mid-range document scores, but bad for the highest scores (in which we are interested in most). Normalisation by the highest computed score can compensate this effect partially, but further improvements are required (e.g., by using a different distribution for the indexing weights).

Although the major advantage of the MIND framework is that is does not *only* address retrieval quality but also other cost factors which are important to a user, one of the next steps should be the comparison of the retrieval quality with other approaches, mainly with CORI [2].

Then, we will study how retrieval quality decreases with other cost factors (time, money). The interesting question is if we achieve faster distributed retrieval (corresponding to communication time by querying less libraries) without a significant decrease of retrieval quality.

Furthermore, we have to extend this framework to other media types besides text (particularly images) and to libraries which do not follow the retrieval model we assumed in this poster.

A fourth aspect of the cost function could be the overlap of digital libraries. If we assume an environment where library documents corresponding to the same object (e.g., two documents referring to the same conference paper, but one document contains bibliographic details and the other the abstract and a link to the full-text) cannot me merged, the user does not want such duplicate documents.

The first idea is to introduce a symmetric matrix D (see 7.1), where

$$D_{ij} := Pr(d \in DL_i \cap DL_j | d \in DL_i \cup dDL_j).$$

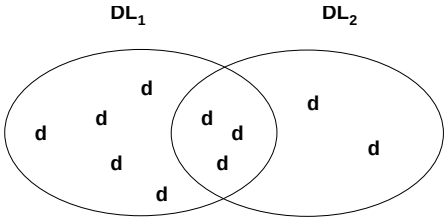


Figure 7.1: DL_1, DL_2 with overlap $D_{12} = D_{21} = Pr(d \in DL_i \cap DL_j | d \in DL_i \cup dDL_j) = 0.3$

Introducing a fourth user-specific mixing parameter $c^O \in [0,1]$, the function $U(\vec{s}, q)$ can now be extended straight forward by an overlap part

$$O(\vec{s}, q) := \frac{2}{m(m-1)} \sum_{i<j} f(s_i) f(s_j) D_{ij} \in [0,1].$$

Obviously, more work has to be spent on this field, especially on methods for estimating the overlap between two databases. We will do so in the next months.

Appendix A

Score density for two terms with exponential distribution

In subsection 4.3.1, an exponential distribution is regarded for the indexing weights.

This appendix shows that the linear combination of exponentially distributed random variables is no exponential distribution.

We assume two exponential distributions with parameters λ_1 and λ_2, respectively. Then, we have the density functions

$$\begin{aligned} p_1(\mathit{weight} = x) &:= \frac{\lambda_1}{1-\exp(-\lambda_1)} \exp(-\lambda_1 x) 1_{[0,1]}(x), \\ p_2(\mathit{weight} = x) &:= \frac{\lambda_2}{1-\exp(-\lambda_2)} \exp(-\lambda_2 x) 1_{[0,1]}(x). \end{aligned}$$

For the linear combination $x := a_1 \cdot x_1 + a_2 \cdot x_2$ of the two exponentially distributed random variable x_1 (with density p_1) and x_2 (with density p_2), we get

$$\begin{aligned} p(x) &= \int_{-\infty}^{\infty} \frac{1}{a_2} p_1(x_1) p_2(\frac{1}{a_2}(x - a_1 \cdot x_1))\ dx_1 \\ &= \int_{-\infty}^{\infty} \frac{1}{a_2} \frac{\lambda_1}{1-\exp(-\lambda_1)} \exp(-\lambda_1 x_1) \frac{\lambda_2}{1-\exp(-\lambda_2)} \exp(-\lambda_2 \frac{x - a_1 x_1}{a_2}) \\ &\qquad 1_{[0,1]}(x_1) 1_{[0,1]}(\frac{x - a_1 x_1}{a_2})\ dx_1 \\ &= \int_{a(a_1,a_2,x)}^{b(a_1,a_2,x)} c' \cdot \exp(-\lambda_1 x_1 - \lambda_2 \frac{x - a_1 x_1}{a_2})\ dx_1 \\ &= \int_{a(a_1,a_2,x)}^{b(a_1,a_2,x)} c' \cdot \exp((-\lambda_1 + \lambda_2 \frac{a_1}{a_2}) x_1) \exp(-\frac{\lambda_2}{a_2} x)\ dx_1 \\ &= c' \cdot \frac{a_2}{-\lambda_1 a_2 + \lambda_2 a_1} \cdot \exp((-\lambda_1 + \lambda_2 \frac{a_1}{a_2}) x_1) \exp(-\frac{\lambda_2}{a_2} x) \ \Big|_{a(a_1,a_2,x)}^{b(a_1,a_2,x)} \\ &= c \cdot \exp((-\lambda_1 + \lambda_2 \frac{a_1}{a_2}) x_1) \exp(-\frac{\lambda_2}{a_2} x) \ \Big|_{a(a_1,a_2,x)}^{b(a_1,a_2,x)} \end{aligned}$$

with

$$\begin{aligned} c' &:= \frac{1}{a_2} \frac{\lambda_1}{1-\exp(-\lambda_1)} \frac{\lambda_2}{1-\exp(-\lambda_2)}. \\ c &:= c' \cdot \frac{a_2}{-\lambda_1 a_2 + \lambda_2 a_1}. \end{aligned}$$

The integrating bounds a and b depend on the values of a_1, a_2 and x. We assume $a_1 \leq a_2$:

1. $x \in [0, a_1] \Longrightarrow a(a_1, a_2, x) := 0,\ b(a_1, a_2, x) := \frac{x}{a_1}$

$$p(x) := c \cdot \left(\exp(-\frac{\lambda_1}{a_1} x) - \exp(-\frac{\lambda_2}{a_2} x) \right).$$

2. $x \in [a_1, a_2] \Longrightarrow a(a_1, a_2, x) := 0,\ b(a_1, a_2, x) := 1$

$$p(x) := c \cdot \left(\exp(-\lambda_1 + \lambda_2 \frac{a_1}{a_2}) - 1 \right) \exp(-\frac{\lambda_2}{a_2} x).$$

3. $x \in [a_2, 1] \Longrightarrow a(a_1, a_2, x) := \frac{x - a_2}{a_1},\ b(a_1, a_2, x) := 1$

$$p(x) := c \cdot \left(\exp(-\lambda_1 + \lambda_2 \frac{a_1}{a_2}) \exp(-\frac{\lambda_2}{a_2} x) - \exp(-\lambda_2 + \lambda_1 \frac{a_2}{a_1}) \exp(-\frac{\lambda_1}{a_1} x) \right).$$

Thus, the resulting distribution p is no exponential distribution.

List of Figures

Bibliography

[1] J. Callan and M. Connell. Query-based sampling of text databases. *ACM Transactions on Information Systems*, 19(2):97–130, 2001.

[2] J. Callan, Z. Lu, and W. Croft. Searching distributed collections with inference networks. In E. Fox, P. Ingwersen, and R. Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29, New York, 1995. ACM.

[3] CMU. 100 collection testbed created by sampling trec123-100-bysource-callan99.v2a. `http://la.lti.cs.cmu.edu/callan/Data/DistribIR/trec123-100-sample300-callan99.v1a.gz`.

[4] CMU. 100 collection testbed created from trec cds 1,2,3. `http://la.lti.cs.cmu.edu/callan/Data/DistribIR/trec123-100-bysource-callan99.v2a.gz`.

[5] S. Fienberg. *The Analysis of Cross-Classified Categorial Data*. MIT Press, Cambridge, Mass., 2. edition, 1980.

[6] D. Freeman. *Applied Categorial Data Analysis*. Dekker, New York, 1987.

[7] N. Fuhr. A decision-theoretic approach to database selection in networked ir. *ACM Transactions on Information Systems*, 17(3):229–249, 1999.

[8] N. Fuhr and U. Pfeifer. Combining model-oriented and description-oriented approaches for probabilistic indexing. In A. Bookstein, Y. Chiaramella, G. Salton, and V. Raghavan, editors, *Proceedings of the Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 46–56, New York, 1991. ACM.

[9] L. Gravano and H. Garcia-Molina. Generalizing gioss to vector-space databases and broker hierarchies. In U. Dayal, P. Gray, and S. Nishio, editors, *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases*, pages 78–89, Los Altos, California, 1995. Morgan Kaufman.

[10] L. Gravano, H. Garcia-Molina, and A. Tomasic. The effectiveness of gloss for the text database discovery problem. In R. T. Snodgrass and M. Winslett, editors, *Proceedings of the 1994 ACM SIGMOD. International Conference on Management of Data.*, pages 126–137, New York, 1994. ACM.

[11] B. Kahle, H. Morris, J. Goldman, T. Erickson, and J. Curran. Interfaces for distributed systems of information servers. *Journal of the American Society for Information Science*, 44(8):453–467, 1993.

[12] R. Manmatha, T. Rath, and F. Feng. Modeling score distributions for combining tzhe outputs of search engines. In *Proceedings SIGIR*, 2001.

[13] S. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33:294–304, 1977.

[14] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, M. Gatford, and A. Payne. Okapi at TREC-4. In *Text REtrieval Conference 4*, pages 73–96, 1996.

[15] J. Schürmann. *Polynomklassifikatoren für die Zeichenerkennung. Ansatz, Adaption, Anwendung.* Oldenbourg, München, Wien, 1977.

[16] H. Turtle and W. Croft. Efficient probabilistic inference for text retrieval. In *Proceedings RIAO 91*, pages 644–661, Paris, France, 1991.

[17] C. J. van Rijsbergen. A non-classical logic for information retrieval. *The Computer Journal*, 29(6):481–485, 1986.

[18] S. Wong and Y. Yao. On modeling information retrieval with probabilistic inference. *ACM Transactions on Information Systems*, 13(1):38–68, 1995.