A selectivity model for fragmented relations: Evaluated for different standard data distributions

Henk Ernst Blok[1] blok@cs.utwente.nl
Sunil Choenni[1,2] s.choenni@nyenrode.nl
Katarzyna Wac[1,3] wac@cs.utwente.nl
Henk M. Blanken[1] blanken@cs.utwente.nl
Peter M.G. Apers[1] apers@cs.utwente.nl

[1]Computer Science Department, University of Twente
PO Box 217, 7500 AE, Enschede, The Netherlands
tel. +31 53 489 3690, fax. +31 53 489 2927
[2]University Nyenrode, Straatweg 25, 3621 BG, Breukelen, The Netherlands
[3]Wroclaw University of Technology, W. Wyspianskiego 27, 50-370 Wroclaw, Poland

Abstract

In the estimation of selectivity, many models assume that data is uniformly distributed, which is not true for many applications. In this paper, we discuss a generalized selectivity model, the so-called $l\alpha\beta$-model which is independent of the data distribution. The model predicts the fraction of a relation that should be selected in order to process a query. We have evaluated this model for different data distributions in order to determine the accuracy of this model. Data distributions that have been considered are the uniform distribution, the normal distribution, the exponential distribution, Pearson's distribution, and Zipf's distribution. From our experiments, it appears that the $l\alpha\beta$-model predicts the selectivity well, especially for the skewed distributions. Applying the $l\alpha\beta$-model on different fragment sizes of a relation yields quite acceptable selectivity values as well.

Keywords: selectivity, fragmentation, databases

1 Introduction

Efficient and effective processing of large amounts of data is of crucial importance in most computer applications, from administrative data processing to library information retrieval systems. Since the first and most important applications were produced in administrative areas, research in efficient and effective processing of data was primarily focussed to meet their performance requirements. These efforts have resulted in query optimizers that perform quite well. An optimizer needs, among others, the selectivity of a query, i.e., the relative number of records that qualifies to a query, in order to generate an efficient query execution plan. The problem of estimating reliable selectivity values has extensively been studied for standard applications under a number of assumptions valid for these applications. For example, many efforts devoted to the selectivity problem assume that data is uniformly distributed, which is not the case for many (emerging) advanced applications. For example, in the field of text retrieval systems, Zipf distribution of data [Zip49] is the norm.

In [BCBA01, BCBA, Blo02], a selectivity model — in the context of information retrieval — is derived that is independent of a particular data distribution. This model, called the $l\alpha\beta$-model, is able to estimate the selectivity of a query in a fast way by means of a mathematically closed formula. Though looking like a so-called parametric model, our model does not depend on a specific distribution function as becomes clear in the next sections. In this paper, we generalize the concepts behind the $l\alpha\beta$-model and report on the accuracy and usefulness of this formula. Generalization of the concepts has as advantage that the model directly can be used for other applications.

We consider a sequence C of lists $L_1, L_2, ...L_{|C|}$ of some arbitrary length $|C|$. An element in a list is called an entity e. In a relation `COLL`, we store pairs (e_i, L_j), indicating that entity e_i is a member of L_j[1,2] . Let

[1]As becomes clear, actually only the e_i column of `COLL` is of interest and the L_j column may be of a different type in practice as well. However, we use this notation for simplicity.

[2]In an information retrieval context, each L_i can be regarded as a document, and an entity as a term. The pair (e_i, L_j) can

a query be defined as the join between an unary relation `Q`, consisting of entities, and `COLL`. The problem is to estimate (beforehand) the size of the join result between `COLL` and `Q`. We have used the $l\alpha\beta$-model to estimate the size of the join result and analyze the accuracy of this model. Note that the size of the join is the number of pairs of `COLL` that satisfy `Q`. The selectivity of `Q` is this number of pairs selected by `Q` in `COLL` relative to the size of `COLL`, i.e., the fraction of `COLL` selected by `Q`.

We have performed a series of experiments for a number of well-known data distributions. The data distributions that we have considered are the uniform distribution, the normal distribution, the exponential distribution, the Pearson distribution, and the Zipf distribution. For each distribution, we ran a group of queries with different length. Then, we measured the selectivity of each query and compared it with the estimated selectivity obtained by applying the $l\alpha\beta$-model. From our experiments, it appears that these two selectivities, the measured ones and the estimated one computed by the $l\alpha\beta$-model, match quite well.

We have also performed these experiments for fragments of different size. Considering different fragments is interesting from a performance point of view (also see Section 2). For example, in the field of information retrieval 80% of the queries can be handled by 20% of the data. If we choose a proper fragment, then we can handle most of the queries with a very limited data set. From our experiments, it turns out that the $l\alpha\beta$-model also works quite well for different fragments, especially for skewed distributions, like the exponential, Pearson, and Zipf distributions.

As in [BCBA01, BCBA, Blo02], we assume that the query and data distributions are known a priori. This assumption appears to be reasonable for a number of applications, such as in the field of information retrieval, data warehousing. Furthermore, the query and data distributions are assumed to be equal and the query is assumed to consist of unique entities.

1.1 Related work

In the literature, a large number of efforts has been reported on the prediction of selectivity in different contexts and under different assumptions [Car75, Yao77, IB86, LNS90, CR94, IP95, GGMS96, PIHS96, CMN98, CMN99]. Roughly two directions can be distinguished in the prediction of selectivities. Research in the first direction has been focussed to the prediction of the number of page or block accesses, to retrieve τ tuples from R tuples which are randomly distributed on B blocks. This problem has been extensively investigated leading from open [Car75] to closed mathematical formulae [Yao77] for predicting the selectivity.

The second research direction mainly focuses on the prediction of intermediate join or selection result sizes. This area has also been subject to research extensively and can be divided into four categories: non-parametric, parametric, curve fitting, and sampling. We refer to [CR94] for a more detailed description of each of these.

The topic of this paper is the evaluation of the $l\alpha\beta$-model. This model fits in the second research direction. However, this model differs on two fundamental points compared to the above-mentioned categories. First, the model allows to estimate the selectivity for a fragmented database. We have extensively evaluated this capability of the model. We note that fragmentation is required for applications, in which the selection of a proper fragment size is crucial (see Section 2 for a number of examples).

Second, the $l\alpha\beta$-model we propose in this paper to estimate the selectivity for fragmented databases, does not fit very well in the categorization typically used in the second research direction. The $l\alpha\beta$-model is not a parametric, sampling, curve fitting, or non-parametric method, or at least not in the usual sense. Our model relies on two parameters that are computed from the data distribution and it can be regarded as a combination of a non-parametric and a parametric approach [BCBA].

Finally, we want to point out that a notion of the costs related to fragmentation might be of use in top-N query optimization [CK98, FSGM+98, CG99, DR99], multi-query optimization [CKSA96], and distributed database query optimization [HKWY97]. The relationship between our research and these topics becomes more clear in Section 2.

be considered as term e_i appears in document L_j.

COLL($ent, entlst$)	(1)
Frequency($ent, freq$)	(2)
Q(ent)	(3)
FragEnt(fno, ent)	(4)

Figure 1: Basic schema definitions.

1.2 Outline

The remainder of this paper is structured as follows. First, we discuss our problem in more detail. Then, we present our selectivity model in Section 3. In Section 4, we discuss a number of standard data distributions. In Section 5 and Section 6, we present the experimental setup and the results that we have obtained, respectively. Finally, Section 7 concludes the paper.

2 Problem statement

The storage of integrated data is rapidly growing, especially in the field of data warehouses. This development supports the progress of a number of advanced applications, such as data mining, decision support systems, multi-media databases. To meet the performance demands of these applications, a widely used strategy is to exploit main-memory capacity by loading the partition of the data in the main memory that is most beneficial.

A similar strategy is applied in the field of information retrieval systems. In these systems, each document is indexed by a large number of terms. All indexed terms might be stored in relation COLL, which is very large. In general it is not efficient to store the whole relation COLL in main memory for several reasons. One reason is that there is not enough space in the main memory. Another reason is that the indexed terms as well as queries on these terms are distributed according to the rule of Zipf, and therefore a relatively small partition of COLL is sufficient to handle the major part of the queries on COLL.

Therefore, a practical need exists for selectivity models that are capable of predicting the selectivity of a query for different data distributions and fragment sizes.

In the following, we organize our discussion in an abstract manner. Let us consider the following four key relations (see Figure 1):

Entity-list pairs [Expr. 1] Each time an entity, *ent*, occurs in a list, *entlst*, an *ent*-*entlst*pair is recorded in the relation COLL. As mentioned before, this relation, which actually is an *inverted list*, is grouped by entities, and then ordered on ascending group count.

List frequencies [Expr. 2] The Frequency relation contains for every entity, *ent*, its list frequency, *freq*. The *freq* of an entity is the number of lists that *ent* occurs in and equals the group count of the *ent* in COLL. This relation can be seen a degenerated histogram, having bin-width 1.

Query [Expr. 3] This relation is nothing more than a set of entities. In practice this relation is constructed by the application or results from another (branch in the same) query expression.

Fragmentation index [Expr. 4] This relation, in practice, serves as an index to find the fragment to which an entity, *ent*, belongs. In our experimental setting, we use this relation for a different purpose as becomes clear in Section 5. For each entity *ent* in *entlst*, this relation contains a tuple $< fno, ent >$, where $fno \in \{1, 2\}$.

In Figure 1, we listed these relations in a more concise form and in Figure 2 the relationships between these relations are depicted.

Let us assume that we are only interested in the m tuples of relation Frequency with the lowest *freq* values and their corresponding tuples in COLL, which is COLL′. Then FragEnt contains a tuple for each of these

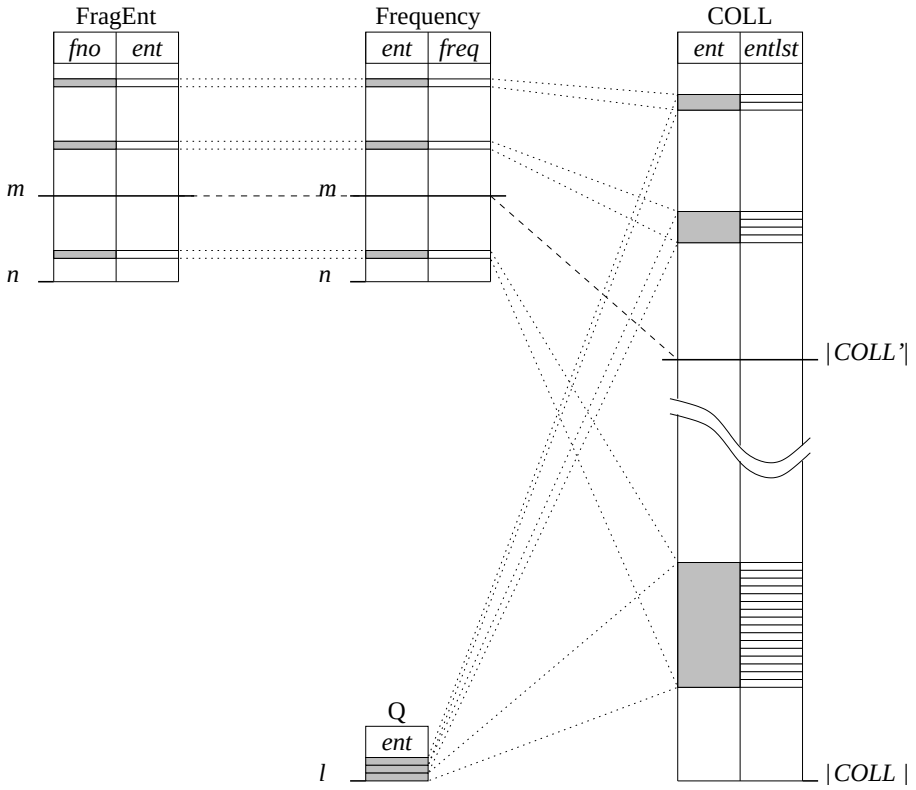


Figure 2: Relations.

entities with $fno = 1$. For all the other entities `FragEnt` has a tuple with the respective entity and $fno = 2$. The $l\alpha\beta$-model is able to predict the size of the (semi)join between `COLL'` and `Q`. We are interested in how good the prediction of this model is. Therefore, we have set up a series of experiments in order to extensively evaluate the accuracy of this model. We vary the data distribution in `COLL` and the length of the query, i.e., the number of entities in `Q`. For several reasons such a prediction is useful.

First of all, when working in main memory `COLL'` should fit into main memory, as well as the result between the (semi)join of `COLL'` and `Q`, during query evaluation time. Therefore, we need to have a notion of the size of that result at design time to be able to determine how much space is left in memory for `COLL'`. A selectivity value helps to determine the size of `COLL'`, or the m value corresponding to it.

Secondly, fragmentation is a tool useful for distributing or parallelizing (shared nothing) databases. Especially in this case, not all the details of the data distribution are available at the global level [HKWY97]. But one still wants to make predictions about the costs, either at design time to distribute the fragments over the nodes, or at run-time to divide the query task over the nodes. So, the used selectivity model preferably does not require the data distribution to be known globally when the model is used.

Thirdly, fragmentation might facilitate top-N query optimization. As proposed in [DR99], one can optimize for top-N queries by guessing a subset of the original data that hopefully suffices to compute the top-N, leaving out the computational effort that otherwise would have been needed to evaluate the ignored data. In particular in the area of information retrieval, top-N queries play an important role: in most cases the top of a ranked list of documents is required. Top-N optimization techniques, therefore, have been subject to extensive research in the information retrieval field. It is quite common to start query evaluation with the terms with the lowest document frequency, being the most discriminative terms[3]. The question should be, both in the information retrieval case as well as in the general case of probabilistic top-N optimization, how to determine the proper size of a subset, e.g., fragment. Cost aspects, and, therefore, a notion of the query selectivity, play an important role in guessing the size of a subset. Of course, a notion of quality plays a role, too. The smaller the fragment used to compute the top-N, the worse the quality will be, which might

[3]Note that in our case terms can be seen as entities.

require additional fragments to be taken into account to reach the desired top-N (see [BVBA01, BHC+01]).

Finally, when dealing in a real world situation the query load can be quite high. This is particularly the case for (new) advanced application areas such as search engines. The ability to handle multiple queries within a short time of each other is therefore important. A high query load often means that some parts of the data are needed quite frequently, since many query results rely on it. Multi-query optimization techniques [CKSA96] exploit this property by trying to reuse (intermediate) query results with regard to a query to speed up evaluation of another. Fragments might be reused as well in this context when more queries rely on it within a short time of each other. A notion of costs, and therefore selectivity, is needed to determine which fragments to reuse or recompute intermediate query results for later.

3 Selectivity model

In this section we reformulate our $l\alpha\beta$-model. As mentioned above, we adopt a more generic notation in this paper in contrast with [BCBA]. This new notation is shown in the Figures 3(a) and 3(b). We use entities [Expr. 5], which may be terms in an information retrieval case, in a certain ordered entity domain [Expr. 6]. We use lists of entities [Expr. 7], which may be documents in an Information Retrieval case. Similar to the documents spanning the term space, these lists span the entity space [Expr. 8]. Note that therefore C is a list covering of E. The set $COLL$ [Expr. 9] is the set wise counterpart of our relation `COLL`.

$$e_i \equiv \text{'an entity'} \quad (5)$$
$$E \equiv [e_1, e_2, e_3, \ldots, e_n] \quad (6)$$
$$L_j \equiv [e_i | e_i \text{ is 'an entity'}] \quad (7)$$
$$C \equiv \left\{ L_1, L_2, L_3, \ldots, L_{|C|} \middle| \bigcup_{j=1}^{|C|} L_j = E \right\} \quad (8)$$
$$COLL \equiv \{(e_i, L_j) | e_i \in L_j \in C\} \quad (9)$$
$$cf_i \equiv |\{(e_i, L_j) | \forall j : e_i \in L_j \in C\}| \quad (10)$$
$$CF \equiv [cf_1, cf_2, cf_3, \ldots, cf_n] \quad (11)$$
$$Q \equiv \{e_i | e_i \in E\},\ |Q| = l \quad (12)$$

(a)

$$E' \equiv \left[e_i \middle| e_i \in E \wedge \forall_{\{e_{i'} \in E \setminus E'\}} cf_i \leq cf_{i'} \right], \quad m = |E'| \quad (13)$$
$$FE \equiv \{(1, E'), (2, E \setminus E')\} \quad (14)$$
$$Q' \equiv \{e_i | e_i \in Q \wedge e_i \in E'\} \quad (15)$$
$$COLL' \equiv \{(e_i, L_j) | e_i \in E' \wedge e_i \in L_j \in C\} \subset COLL \quad (16)$$
$$COLL'_Q \equiv \{(e_i, L_j) | e_i \in Q' \wedge e_i \in L_j \in C\} \subset COLL' \quad (17)$$
$$CF' \equiv [cf_i | e_i \in E'] \quad (18)$$

(b)

Figure 3: Basic mathematical definitions.

For each of the entities e_i, we distinguish the number of lists L_j in which it occurs. We call this the *covering frequency* cf_i [Expr. 10] of e_i. The list CF [Expr. 11] is the counterpart of our (histogram) table `Frequency`. The relation `Q` is modeled as the set Q [Expr. 12].

As we explained before, we are only interested in the m entities with the lowest covering frequencies, being the most restrictive ones in $COLL$. We define the subset E' of E, containing only those entities [Expr. 13]. The set FE contains two pairs corresponding to the two fragments similar to the `FragEnt` relation but now expressed formally in a nested manner [Expr. 14]. In terms of FE, we are thus only interested in fragment 1 denoted by the pair $(1, E')$. We can also define Q' as the version of Q restricted to those entities [Expr. 15]. $COLL'$ [Expr. 16] represents the fragment `COLL'` of `COLL` restricted to these least frequent entities and $COLL'_Q$ [Expr. 17] represents the set wise counterpart of the semijoin between `COLL'` and `Q`. Finally, CF' [Expr. 18] denotes the list of covering frequencies corresponding to the entities of interest.

Using these definitions the actual (measured) selectivity is defined as shown in Figure 4. Our selectivity estimator, see Figure 5, is defined as the product of the query length l and two factors: α [Figure 6, Expr. 21] and β [Figure 6, Expr. 22]. The α can be interpreted as the conditional expected value of the fraction of $COLL'$ selected by an arbitrary entity in Q given that the entity in question is known to be in E'. The β is the probability that an entity in Q is in E'.

$$sel_{measured} \equiv \frac{|COLL'_Q|}{|COLL'|} \tag{19}$$

Figure 4: Measured selectivity definition.

$$sel_{estimated} \equiv l\alpha\beta \tag{20}$$

Figure 5: Estimated selectivity definition.

4 Data distributions

In this section, we describe the five distributions for which we have evaluated our $l\alpha\beta$-model: uniform distribution, normal distribution, exponential distribution, Pearson distribution, and Zipf distribution. Each of these distributions is well-known in the field of selectivity estimation. We describe each of them and stress some practical problems one has to take care of when using the distribution functions to sample data sets.

We refer to Figure 7 for an overview of the distribution density functions $f(x)$. In the Figures 8 to 12 we plotted the distribution density $f(x)$ and the corresponding cumulative distribution $F(x)$ for each of the distributions.

We chose the parameters[4] such that each distribution spans its entire range on the x-domain $[0, 1]$. For those distribution density functions having the x-axis as horizontal asymptote for $x \to \infty$, we chose the parameters such that this asymptote is reached within the numerical precision of our experimental system, i.e., $f(x) < \textit{numerical precision}$ for $x \geq 1$, so numerically $f(1) = 0$. For the distributions with the x-axis as horizontal asymptote for $x \to -\infty$ we followed a similar approach to achieve that numerically $f(0) = 0$. Since, we plan to use these distributions for sampling a very large database we expect this approach to be valid.

Based on the knowledge that the normal distribution is symmetrical, we chose $\mu = 0.5$. The σ was then determined by numerically finding the left and right horizontal asymptote as we described above. A similar approach was used for the exponential distribution, using $\mu = 0$.

For the Pearson distribution the parameters a, b, and p determine the location and the slopes of its peak. These can be chosen arbitrarily within certain limits. Since, we are mainly interested in highly skewed distributions — flat distributions we consider as rather easy, and therefore less interesting, to estimate the selectivity for — we chose a, b, and p such that $f(x)$ indeed is highly skewed. Furthermore, we scaled the distribution such that numerically it reaches its asymptote to make sure that the distribution occupied its entire range within the $[0, 1]$ domain.

The Zipf distribution is a special case, since we used a Zipfian distributed real world data set in [BCBA] and since it has some particular numerical properties. For clarity we start with describing the standard Zipf distribution before we switch to the variant we used in our experiments. Note that Figure 12 shows our variant and not the standard Zipf distribution. The standard Zipf distribution density is $f(x) = \frac{1}{x^p}$ where

[4]Note that in many cases the a-parameter is called α and the b-parameter β in the distribution functions. Since, we already use α and β in our model with a completely different meaning, we use a and b instead to avoid ambiguity.

$$\alpha \equiv \sum_{cf_i \in CF'} \frac{cf_i^2}{\left(\sum_{cf_i \in CF'} cf_i\right)^2} = \frac{\sum_{cf_i \in CF'} cf_i^2}{|COLL'|^2} \tag{21}$$

$$\beta \equiv \sum_{cf_i \in CF'} \frac{cf_i}{\sum_{cf_i \in CF} cf_i} = \frac{|COLL'|}{|COLL|} \tag{22}$$

Figure 6: Additional definitions.

Uniform distribution:

$$f(x) \equiv 1 \tag{23}$$

Normal distribution:

$$f(x) \equiv \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}, \text{ where: } \mu = 0.5,\ \sigma = 0.0875771730212624 \tag{24}$$

Exponential distribution:

$$f(x) \equiv \frac{1}{b} e^{\frac{\mu - x}{b}}, \text{ where: } \mu = 0,\ b = 0.051125 \tag{25}$$

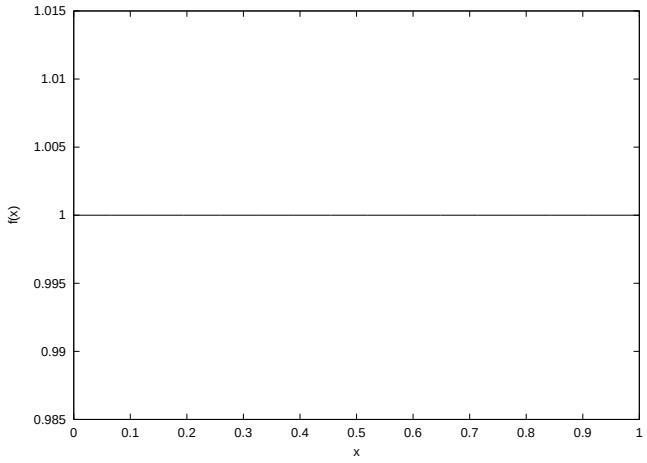
Pearson (type III) distribution:

$$f(x) \equiv \frac{1}{b\Gamma(p)} \left(\frac{x-a}{b}\right)^{p-1} e^{\left(-\frac{x-a}{b}\right)}, \text{ where: } p = 4,\ a = 0,\ b = 0.0075 \tag{26}$$

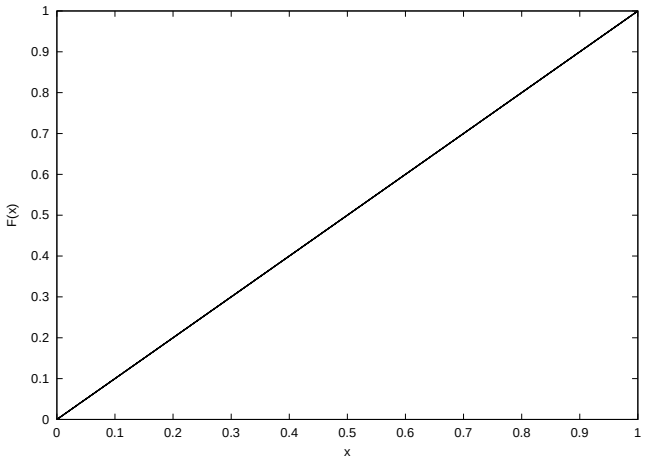
Zipf distribution:

$$f(x) \equiv \frac{a}{b-x} + c, \text{ where: } a = 0.169183259456481,\ b = 1.00099900199501,\ c = -0.169014413719988 \tag{27}$$

Figure 7: Distribution density functions.

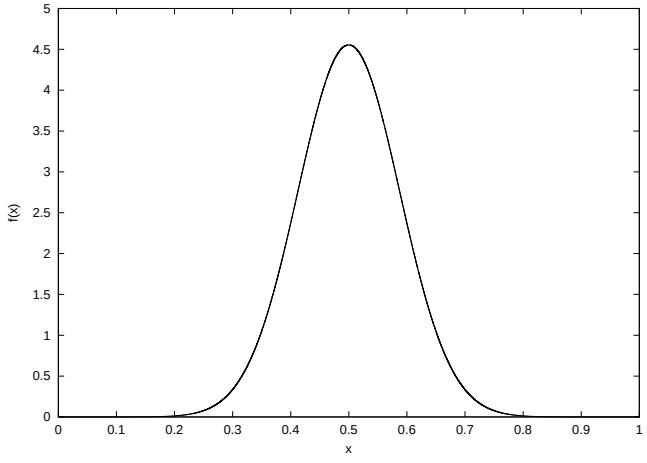


(a) Distribution density.

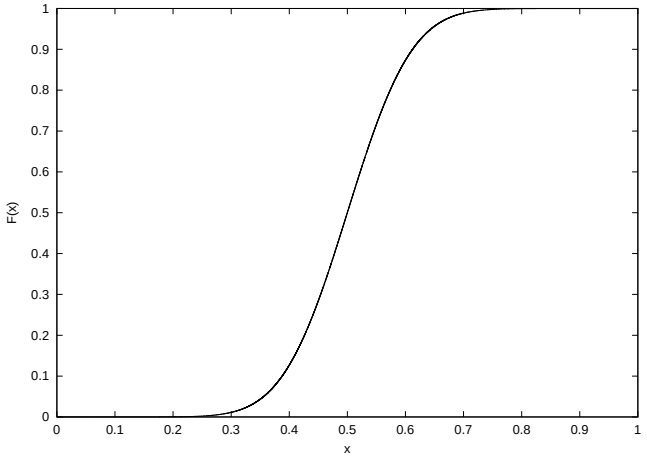


(b) Cumulative distribution.

Figure 8: Uniform distribution.

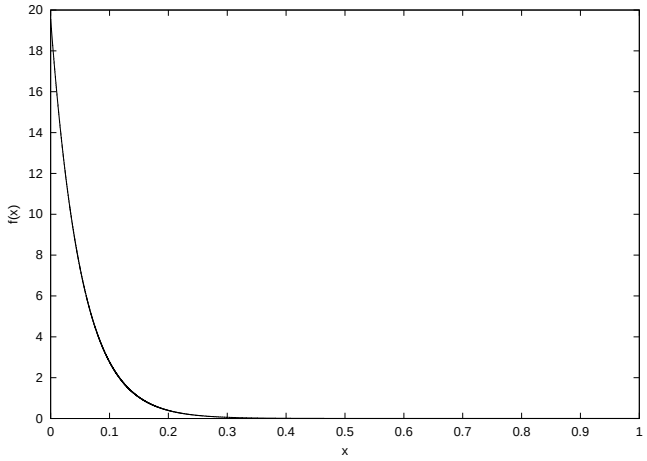


(a) Distribution density.

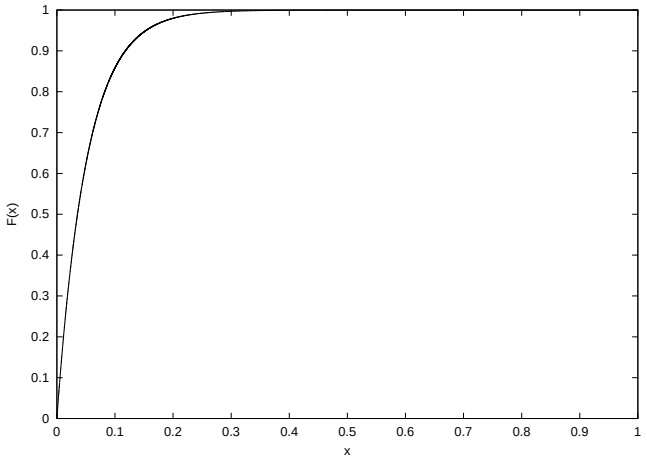


(b) Cumulative distribution.

Figure 9: Normal distribution.

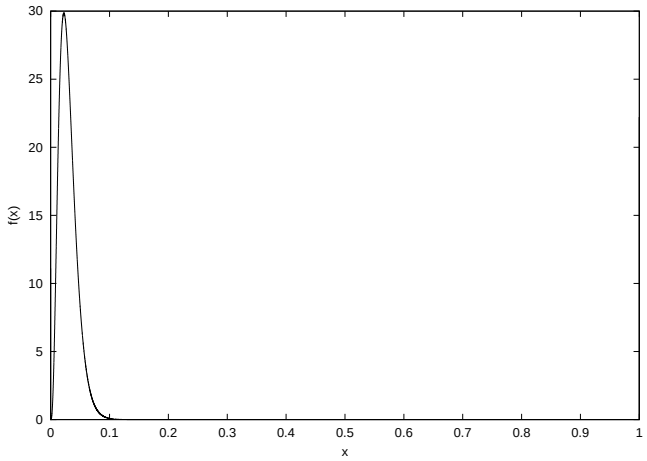


(a) Distribution density.

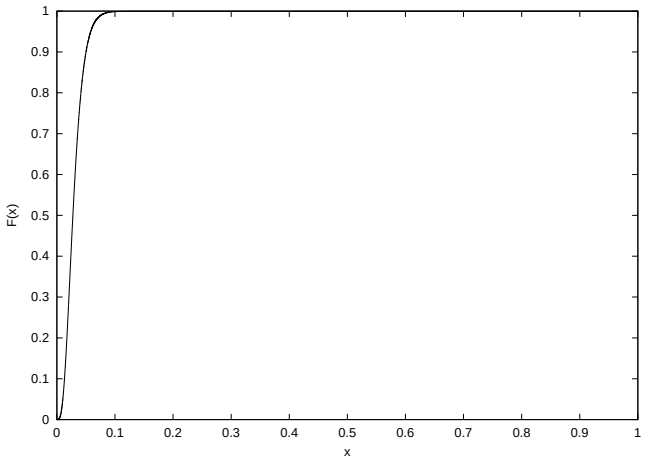


(b) Cumulative distribution.

Figure 10: Exponential distribution.

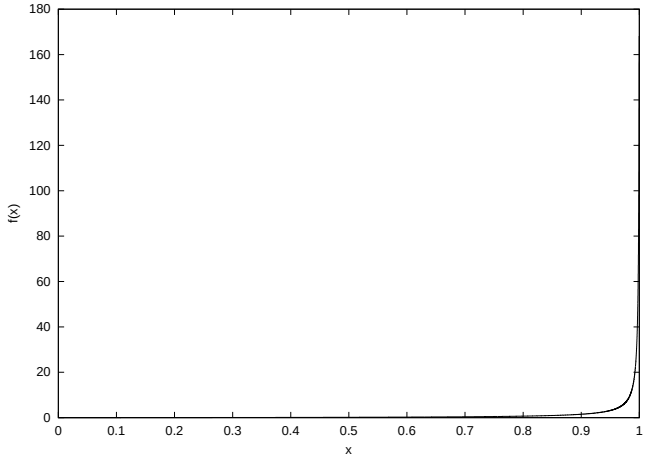


(a) Distribution density.

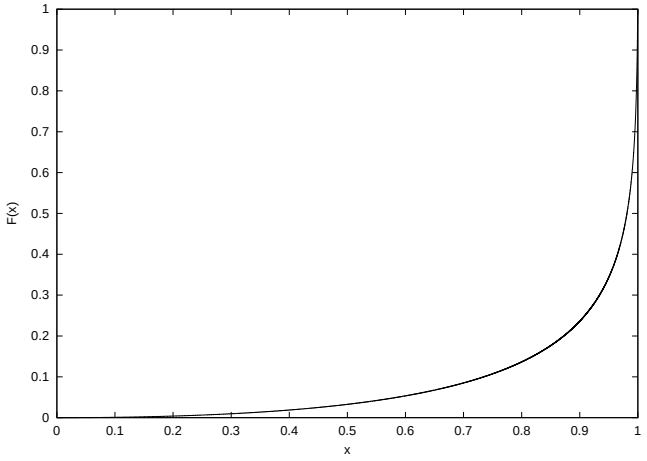


(b) Cumulative distribution.

Figure 11: Pearson distribution.



(a) Distribution density.



(b) Cumulative distribution.

Figure 12: Zipf distribution.

$p \approx 1$. This function has the y-axis as a vertical asymptote next to the x-axis as horizontal asymptote. Furthermore, for $p = 1$ its $F(x)$ is not properly defined as it goes to ∞. Also, $f(x)$ closes in on its asymptotes much slower than the other distributions we are interested in. In [BCBA] we used a mirrored Zipfian distribution for convenience sake, i.e., we switched the left and right hand side. To allow for better comparison with [BCBA] we use a mirrored $f(x)$ in this paper as well. To overcome the numerical problems with the asymptotes and to guarantee the existence of $F(x)$ we also slightly shifted $f(x)$ over a small distance, both vertically as well as horizontally[5]. Requiring that $f(0) = 0$ and $F(1) = 1$ we get two equations with three unknown variables. This leaves one of the three free to be chosen arbitrarily, though within certain limits. Choosing this third parameter determines how skewed the distribution becomes. As for the Pearson distribution, we chose it to obtain a very skewed distribution. However, in this case we let our choice also be inspired by the skewedness of the real world, i.e., TREC [TRE], data sets as used in [BCBA] to allow better comparison of the results. As for the horizontally scaling of the normal, exponential, and Pearson distributions, we consider this approach to fix the numerical problems with the Zipf distribution as valid due to the large numbers planned to be involved when sampling our test database.

5 Experimental setup

In order to perform our experiments, we have chosen the MySQL (version 3.23.49a) database environment as platform in combination with Perl (version 5.005_03) running on a PC running Linux (version 2.2.16-3 #1 SMP).

Our experimental setup is generic of nature and consists of six steps as shown in Figure 13. We first define a database scheme, which is the same for each data distribution. As second, we generate the content of the database according to the characteristics of a given data distribution. Then, the third step generates a set of queries, and the fourth step the distinguished fragmentation. Then, our program performs the semi-join between the first fragment and the queries, followed by the step that computes the estimated and actual selectivity. Finally, these results are aggregated and error statistics are computed. We run this procedure for each of the five distributions of interest as described in Section 4.

The database corresponding to each distribution contains 100000 entities. The query lengths range from 5 to 50 by steps of 5. For each query length, we sample 50 random queries using the same distribution as the data as represented by `Frequency`. We use the Monte Carlo method [HATB98] as sampling method. The relative fragment $\frac{m}{n}$ sizes that we considered are 0.05, 0.10, 0.15, ..., 0.95, and 1.00.

The relative error we compute in the sixth step is defined as follows.

Definition 1 (Relative error) *The relative error $\epsilon_{\overline{x}}$ of an estimated (selectivity) value $\overline{x}$ of a variable x, i.e., a measured selectivity, is defined as:*

$$\epsilon_{\overline{x}} \equiv \frac{\overline{x} - x}{x}$$

for $x \neq 0$.

We want to stress that the relative error is not defined for $x = 0$.

6 Experimental results

In this section, we present the results of the experiments we did as described in the previous section.

For each of the five distributions of interest, we plotted the estimated vs. the measured selectivity in the Figures 15(a), 16(a), 17(a), 18(a), and 19(a). In the remainder of this section we refer to these figures as

[5]Note that we choose $p = 1$. This makes the function easier to understand without a significant impact on the results since p would have been close to 1 anyway.

[Step 1] Initialization

Create a database with the following relations:

`Frequency`$(ent, freq, cumfreqstart, cumfreqstop)$

`FragEnt`(fno, ent)

`QSet`(qno, ent)

`SET` $m := 0.05 \cdot n$

Remarks:

- We added two extra columns to `Frequency`, where for $ent = e_1$ holds $cumfreqstart = 0$, for $entity = e_n$ holds $cumfreqstop = 1$, and for all cases $cumfreqstop = cumfreqstart + freq$.
- We use a binary relation `QSet` instead of `Q` to model a set of queries `Q`, where $qno \in \{1, 2, 3, \ldots, 500\}$.
- The maximum qno of 500 follows directly from the fact that we have 50 queries of each length and we have 10 different lengths, i.e, 5, 10, 15, . . ., 45, and 50.

[Step 2] Generate histogram

Using the formula for $f(x)$ for the distribution of interest we compute the values in the `Frequency` table using numerical integration.

[Step 3] Generate queries

Via Monte Carlo sampling [HATB98], using the two (additional) last columns of `Frequency`, we fill `QSet`.

Remarks:

- $|\texttt{QSet}| = 50 \cdot (5 + 10 + 15 + \ldots + 45 + 50) = 50 \cdot 275 = 13750$, since we have 50 queries of each length and we have as many as the query length is large tuples per query.
- We applied the Monte Carlo method iteratively to remove any duplicate entities within queries, since our model requires that no duplicate entities exist in the queries.

[Step 4] Generate fragmentation

Based on the frequencies in `Frequency` and a given ratio $\frac{m}{n}$ we determine `FragEnt`.

Remarks:

- We only generate the entries for $fno = 1$, since we are only interested in the first fragment.
- We do not actually fragment the histogram but only construct `FragEnt` and we generate fragments on the fly when needed via joining with `FragEnt`.

[Step 5] Compute selectivity

Compute the results:

[a] Compute $sel_{estimated}$ per query.

[b] Compute $sel_{measured}$ per query via joining `QSet`, `FragEnt`, and `Frequency`, group by qno — and, strictly speaking, fno, but we know we only have one fno value, so grouping on fno is superfluous — followed by summing the $freq$ values per group.

Store the results for this fragmentation.

`IF NOT(`$\frac{m}{n} = 1$`) THEN`
  `SET` $m := m + 0.05 \cdot n$
  `GO TO` **Step 4**
`END IF`

Remark:

- Since, the $freq$-values in `Frequency` are normalized to sum to 1 the computations of $sel_{estimated}$ are slightly different than as specified in Expr. 21 and 22 to adjust for this.

[Step 6] Aggregate results

Gather the results and compute the error and relative error for all fragmentations.

Generate an estimated vs. measured selectivity plot and a relative error vs. relative fragment size, i.e., $\frac{m}{n}$, plot.

Figure 13: Experimental process.

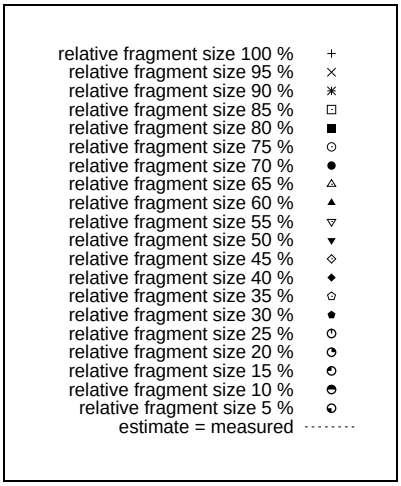


Figure 14: Legend of Figures 15(a), 16(a), 17(a), 18(a), and 19(a).

the (a) plots. We also plotted the ideal line for each of the distribution. This line represents the case where the estimated selectivity value equals the measured selectivity value. If our selectivity model were perfect, all points would be on this line. Note that each fragmentation has its own point markers. For a legend explaining which marker belongs to which fragmentation we refer to Figure 14[6].

To give an impression of the effect of the fragment size on the accuracy of our $l\alpha\beta$-selectivity model we also plotted the relative error vs. the relative fragment size, i.e., $\frac{m}{n}$, for each of the distributions in the Figures 15(b), 16(b), 17(b), 18(b), and 19(b). In the remainder of this section we refer to these figures as the (b) plots. Note that the relative error is only defined for those cases where the measured selectivity value is not equal to 0.

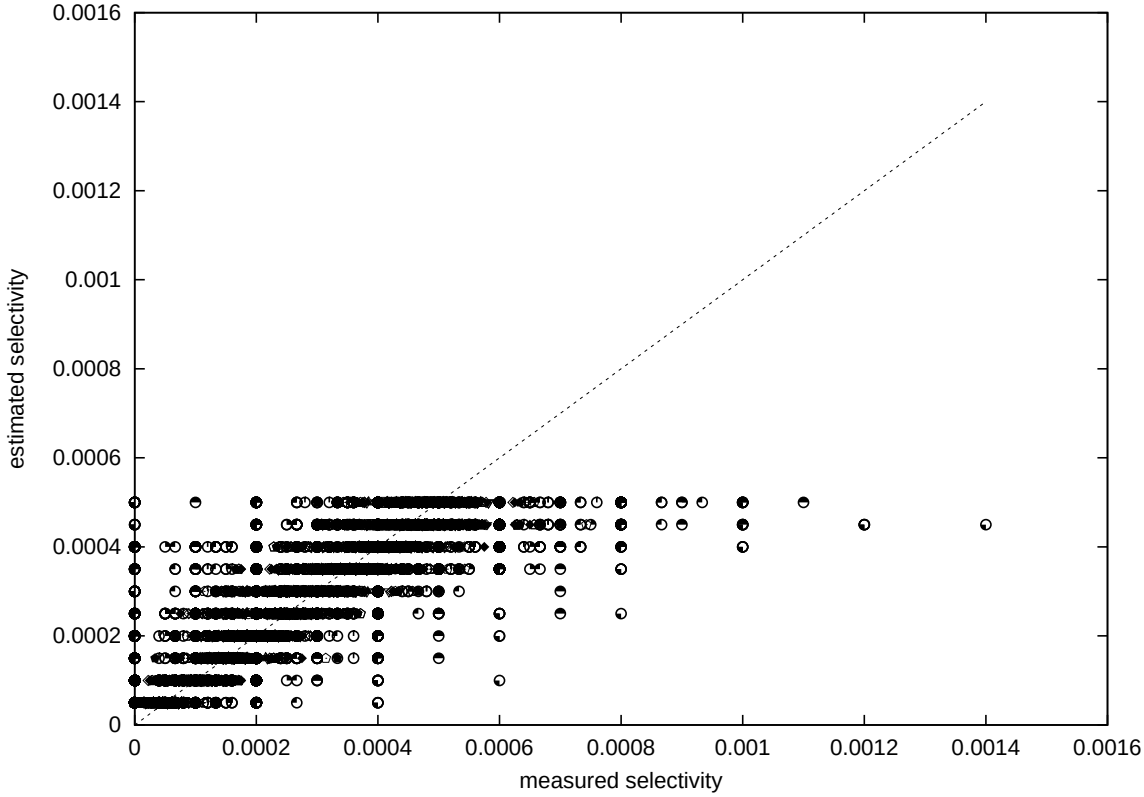
The remainder of this section is dedicated to the discussion of the results.

When having a first glance at the results shown in the (a) plots one notices directly that the $l\alpha\beta$ estimated selectivity indeed represents a good average of the actual selectivities. Looking at the (b) plots one sees, however, that this is not entirely true, since many of the relative errors are positive. This means that in many cases the $l\alpha\beta$-model overestimates the selectivity value. This is not very surprising when we look at the construction of the $l\alpha\beta$-model in [BCBA]. It shows that the $l\alpha\beta$-model is an upper bound for the actual expected value of the selectivity. Also the larger relative errors mainly occur for the larger relative fragments sizes. A further analysis of the results learned us that the larger relative errors mainly occur for the smaller selectivities. This means that the error in absolute numbers is not so large after all. The only exception to this observation is the uniform distribution. This corresponds quite well with what we expected.

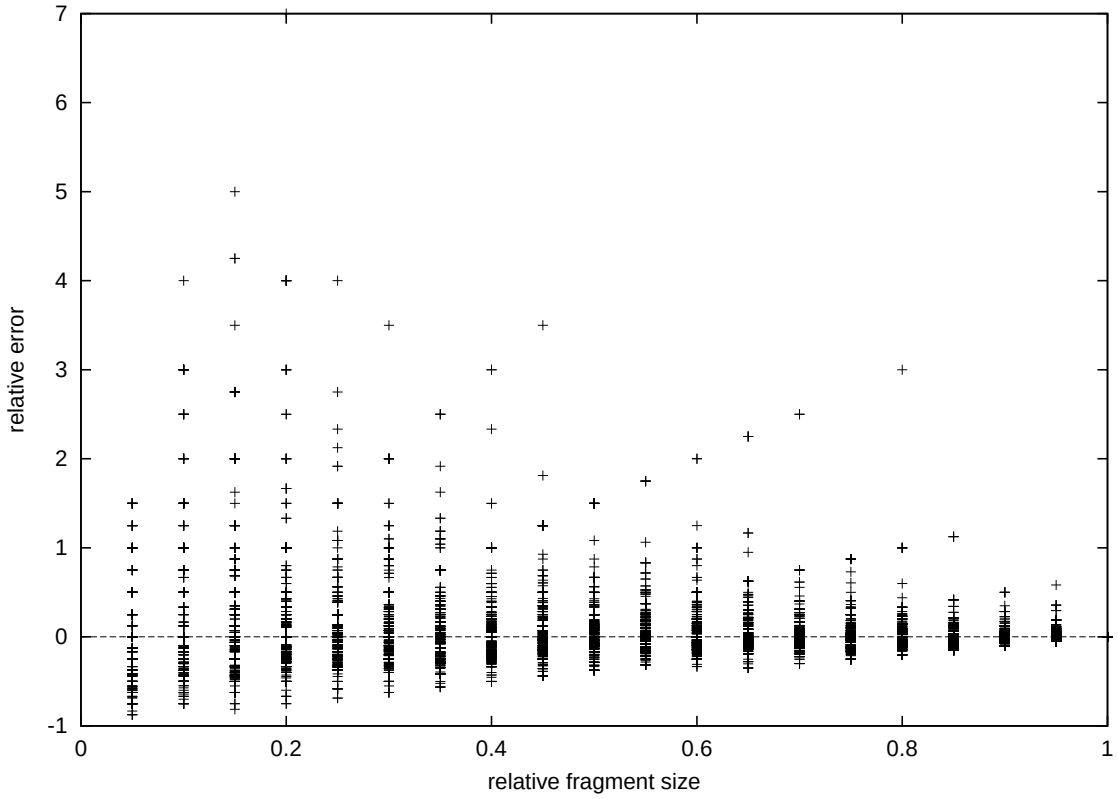
For all distributions, except the uniform, the peak in the distribution is only captured by the fragment of interest for the larger fragment sizes. For increasing fragment size, the amount of data that `Q` is matched against grows rapidly resulting in lower selectivity values since the $|COLL'|$ part in Expr. 19 increases. The larger the part of the peak that is included in the fragment, the bigger the chances that excess selectivities occur resulting in increasing chances for larger estimation errors. Note, however, that for the special case where the fragment `COLL'` equals the entire relation `COLL`, the relative error is about 0 for all distributions.

For smaller fragments of interest and the uniform distribution, the distribution is flat or nearly flat. This means that for decreasing fragment sizes chances increase that an entity in `Q` is not in `COLL'` resulting in increasing chances that entities do not contribute to the measured selectivity. In turn, this results in increasing chances for estimation errors. A closer look at the log files learned that, in particular for the smallest fragments, the probability of entities not being in the fragment is so large that the measured selectivity value becomes very small. So also here, the largest relative errors occurring for the smallest selectivity values appears to be logical.

[6]Note that due to the number of data points most of the different symbols cannot be distinguished in the (a) plots. However, since some still can be distinguished we found it better to present the plots instead of using a single point symbol for all fragmentations.

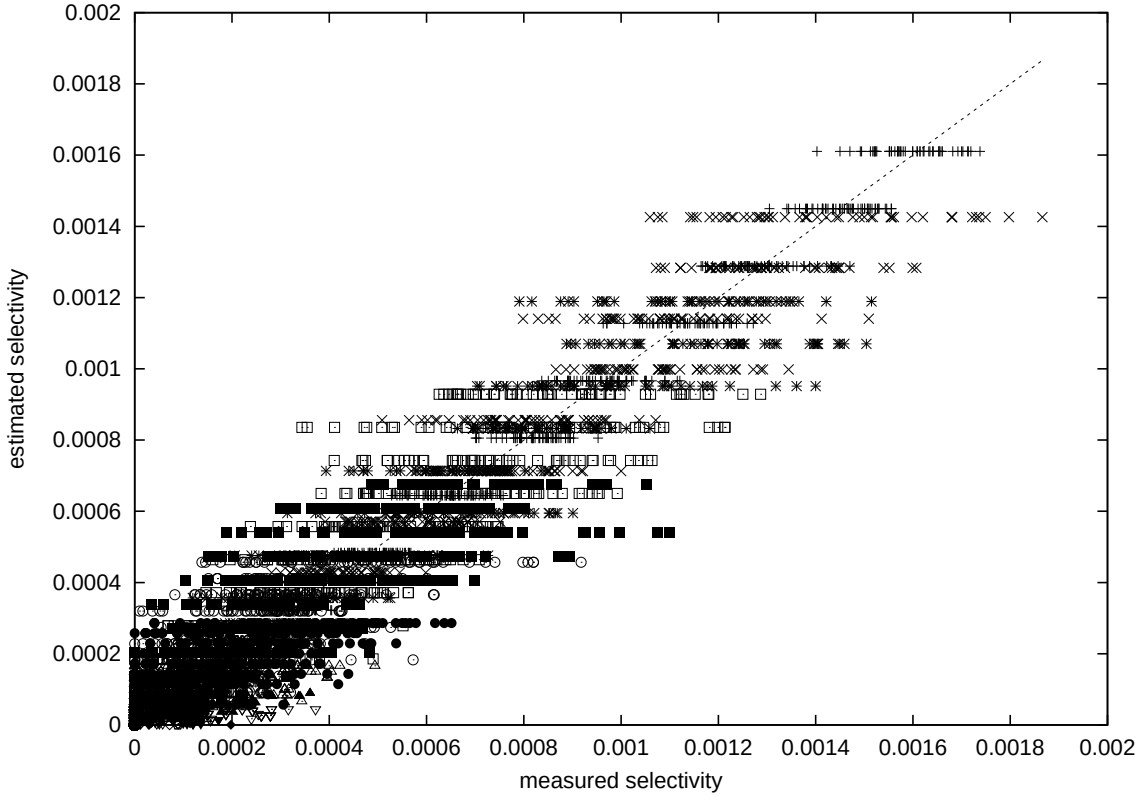


(a) Estimated vs. measured selectivity.

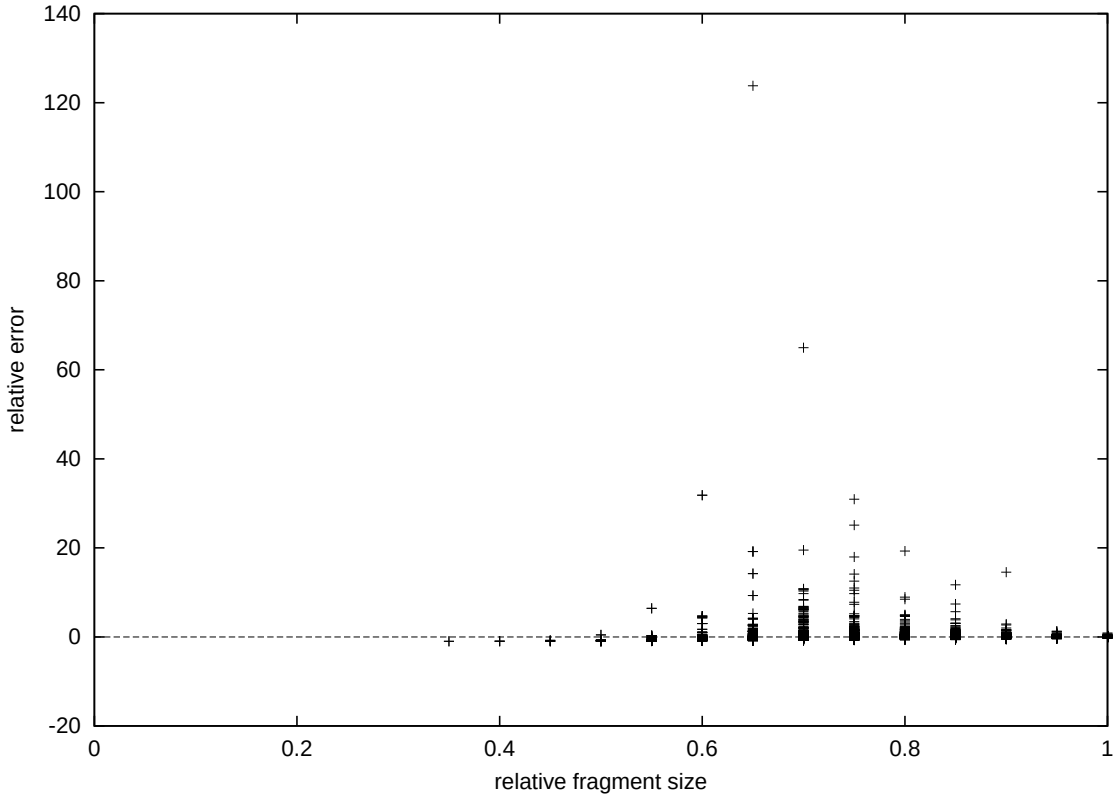


(b) Relative error vs. relative fragment size.

Figure 15: Uniform distribution.

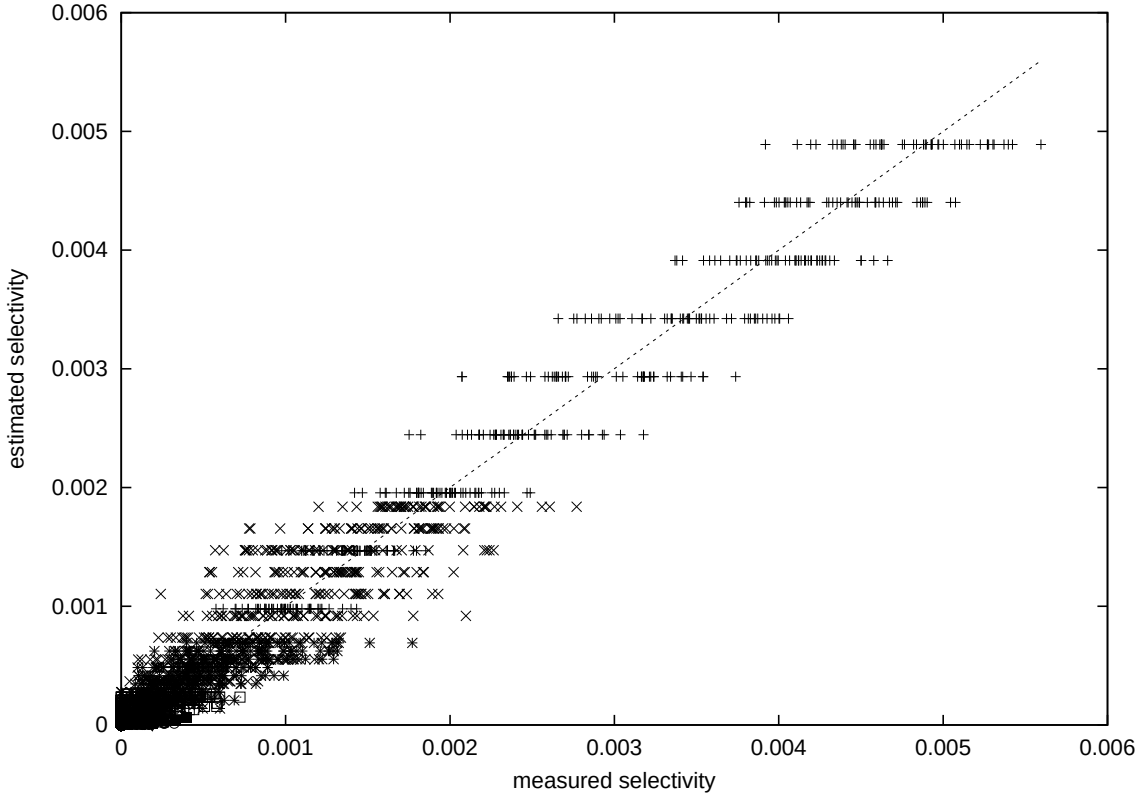


(a) Estimated vs. measured selectivity.

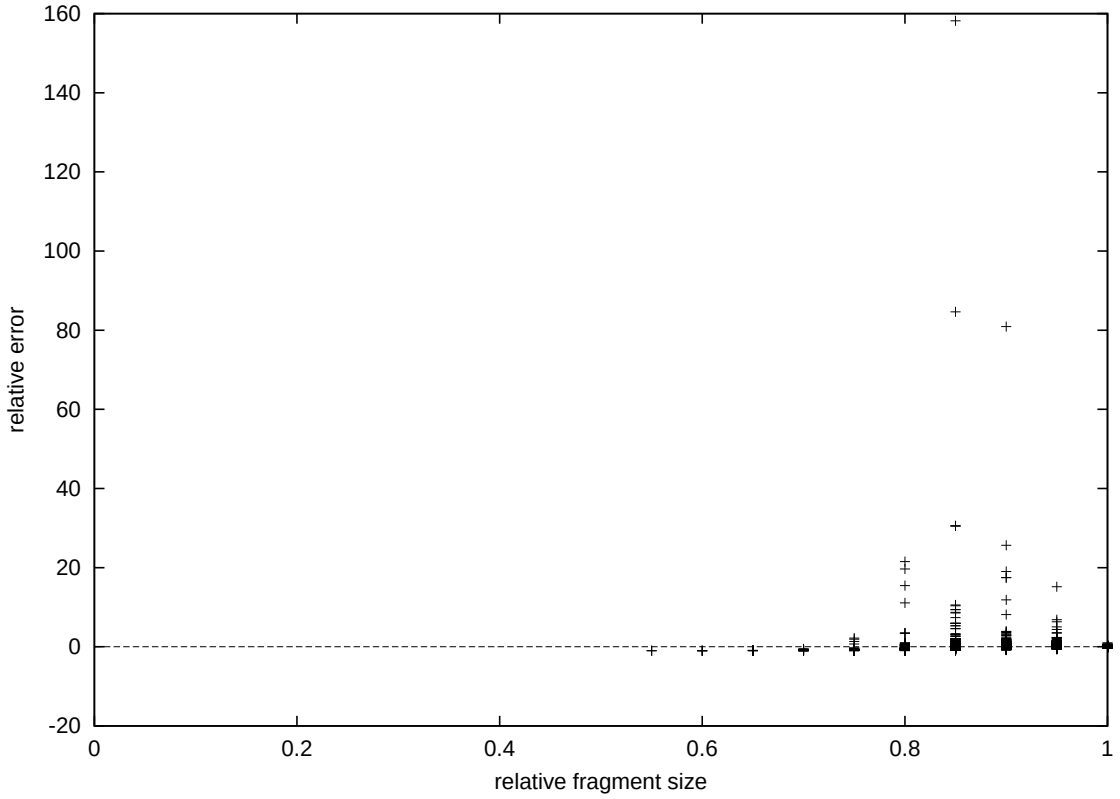


(b) Relative error vs. relative fragment size.

Figure 16: Normal distribution.

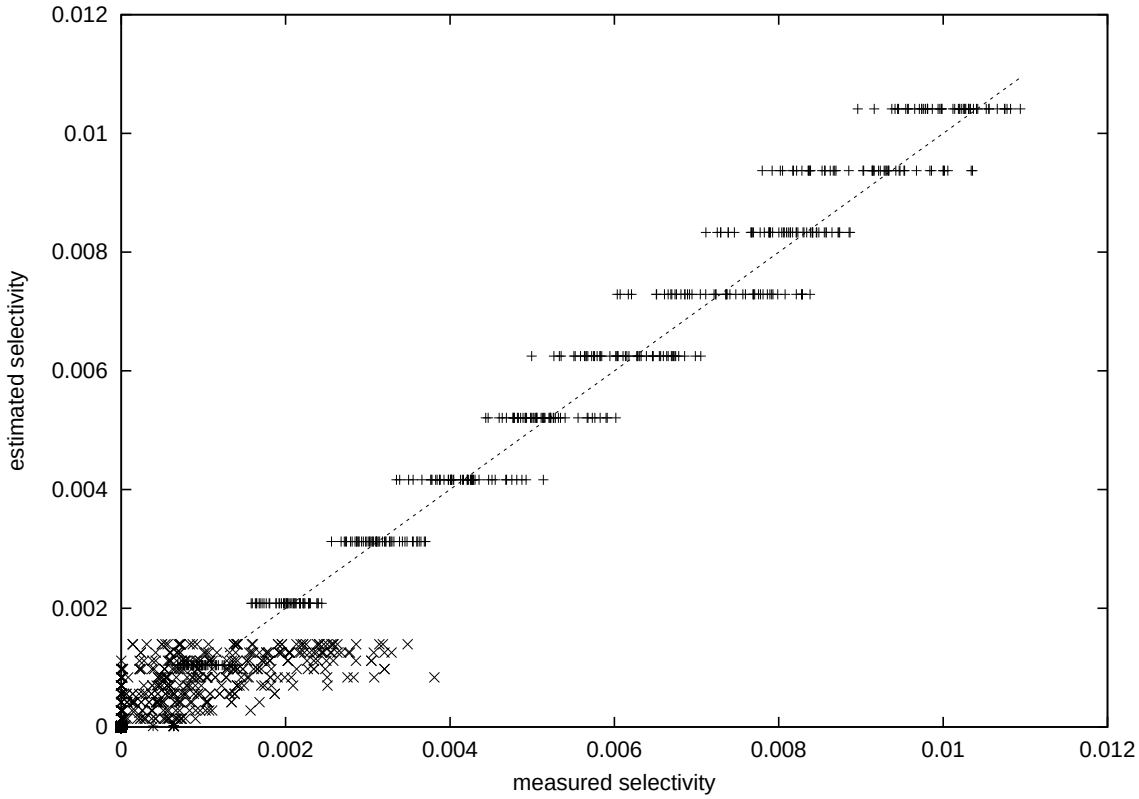


(a) Estimated vs. measured selectivity.

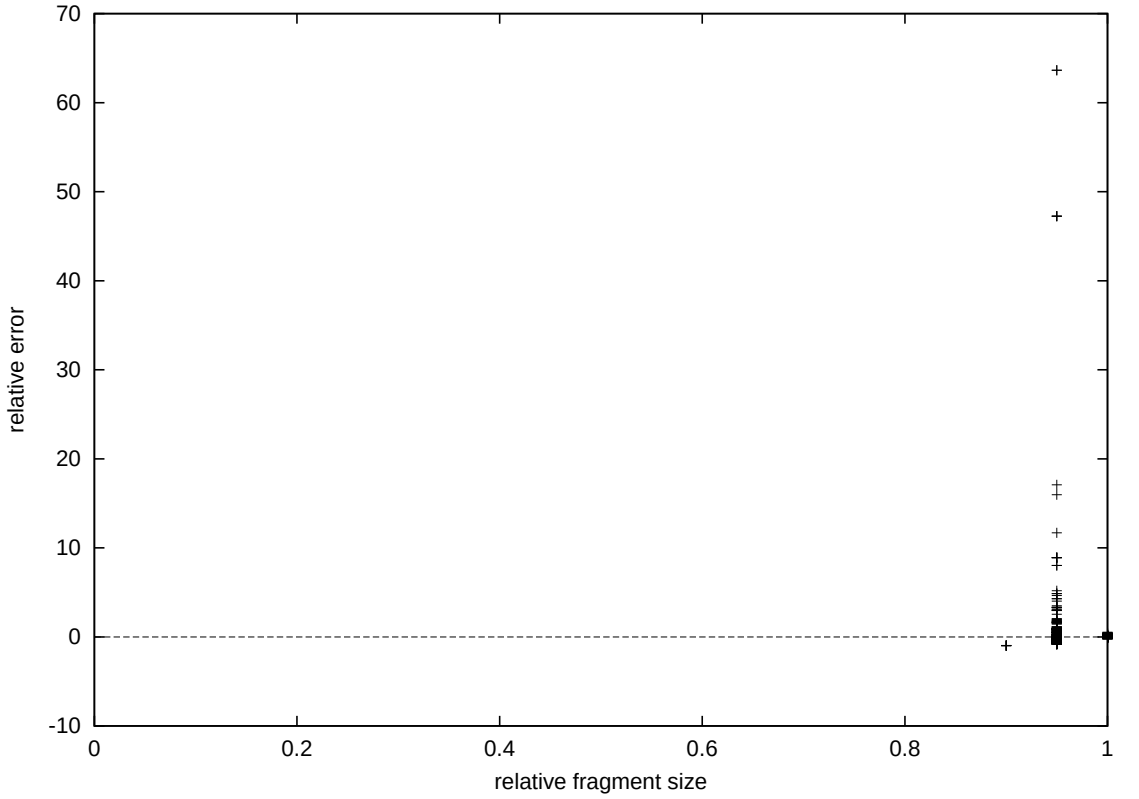


(b) Relative error vs. relative fragment size.

Figure 17: Exponential distribution.

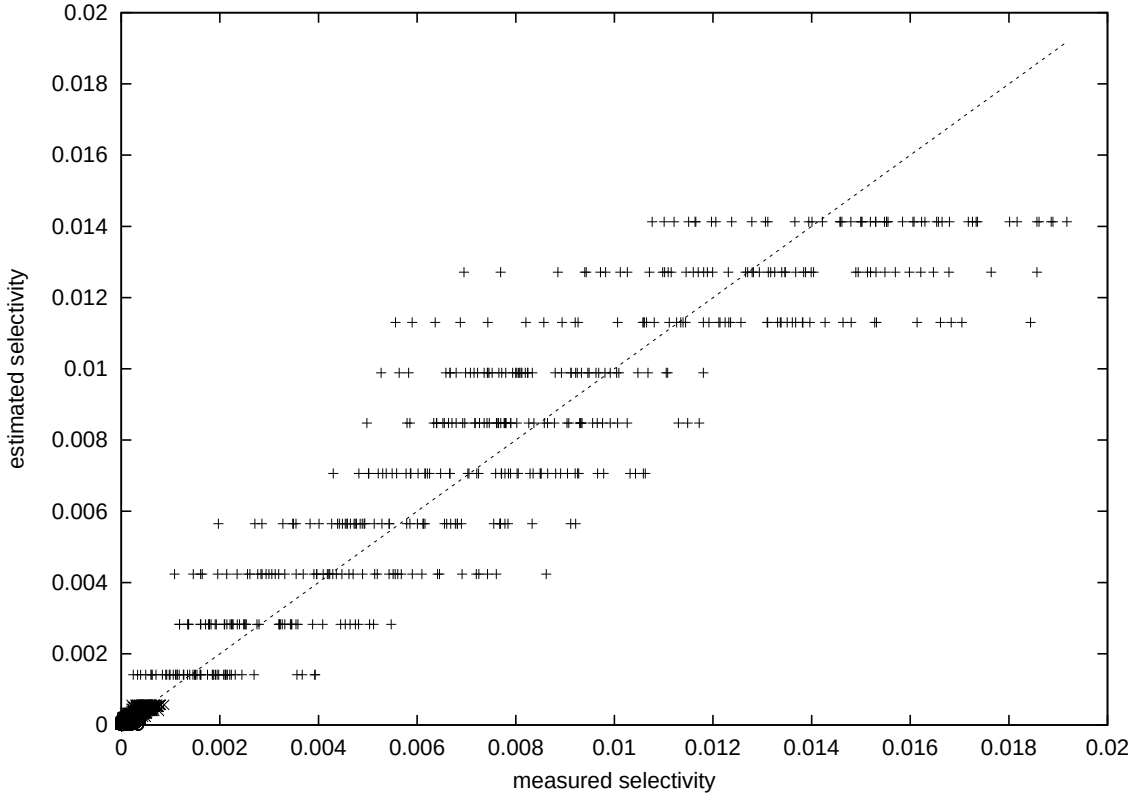


(a) Estimated vs. measured selectivity.

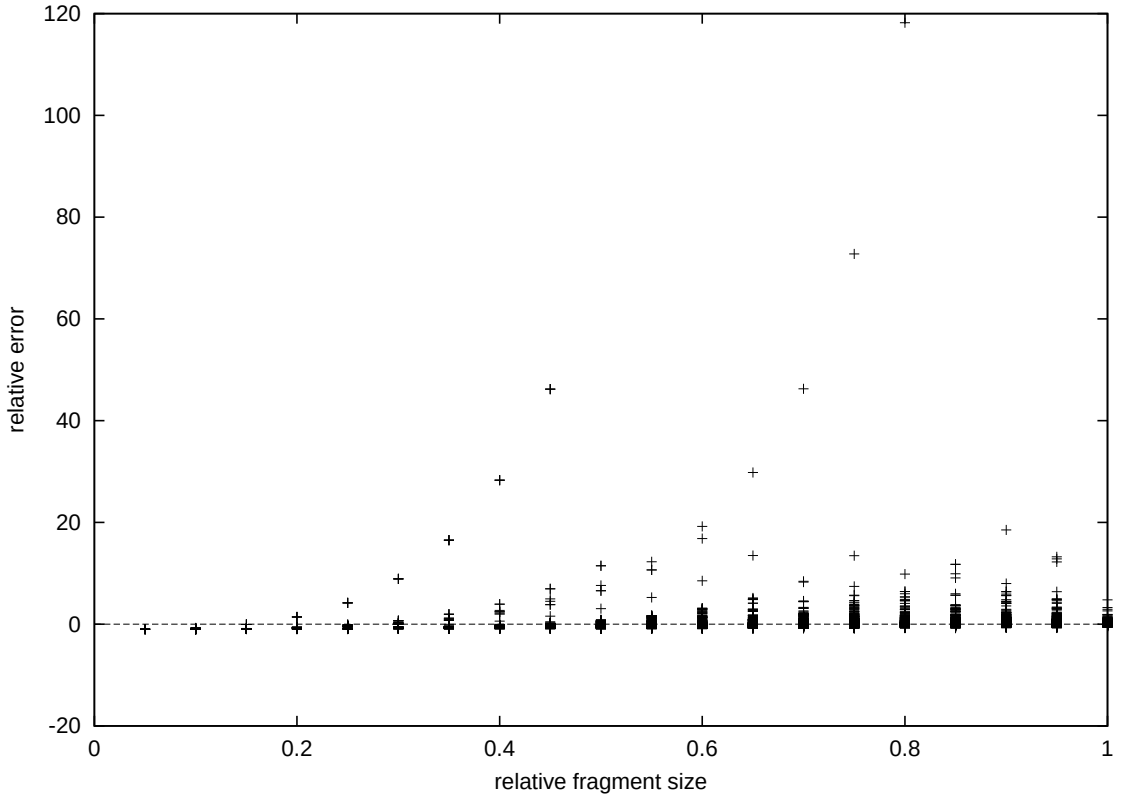


(b) Relative error vs. relative fragment size.

Figure 18: Pearson distribution.



(a) Estimated vs. measured selectivity.



(b) Relative error vs. relative fragment size.

Figure 19: Zipf distribution.

Finally, we have some closing remarks regarding the effectiveness of our $l\alpha\beta$-model. The errors might *seem* very large for a selectivity model, but we think this is acceptable given that these errors are *not so large in absolute terms* (as we explained) and given that our model only uses very little information when estimating the selectivity which has interesting performance benefits as explained in first two sections of this paper. Furthermore, note that our $l\alpha\beta$-model is not just another parametric model since it takes fragmentation into account and also does not presume any (parameterized) standard distribution function. As shown, our $l\alpha\beta$-model performs very well for the unfragmented case as is obvious from the relative error being approximately 0 for this case for all distributions.

7 Conclusions and future research

It is widely recognized that a good estimation of the selectivity of a query is of crucial importance for query processing. Therefore, a lot of research has been devoted towards the prediction of selectivity values, resulting in different selectivity models. However, most of these models assume that data is uniformly distributed and/or are focused towards a specific application. In many (emerging) advanced applications the assumption that data is uniformly distributed does not hold, e.g., in the field of information retrieval data is distributed according to Zipf's law.

In this paper, we generalized the so-called $l\alpha\beta$ selectivity model. This model claims to be independent of a specific data distribution. Therefore, we pose ourselves the question how accurate is the $l\alpha\beta$-model for different types of data distribution. The selectivity is defined as the fraction of a relation `COLL` that is selected by another relation `Q`. We consider five types of well-known data distributions namely, the uniform distribution, the normal distribution, the exponential distribution, Pearson's distribution, and Zipf's distribution. For each data distribution, we have ran different sets of queries and compared the selectivity obtained by applying the $l\alpha\beta$-model with the selectivity that we have measured. Furthermore, we have ran these sets of queries also against different fragment sizes. Our overall conclusion is that that the selectivity values obtained by applying the $l\alpha\beta$-model meets the measured values well. Especially, for the skewed distributions the $l\alpha\beta$-model yields good results. Therefore, the $l\alpha\beta$-model is an accurate model that can predict the selectivity in a quite cheap way. The reader might argue that the errors might seem very large for a selectivity model. However, as we explained in the previous section, we think this is acceptable given that these errors are not so large in absolute terms and given that our model only uses very little information when estimating the selectivity which has interesting performance benefits as explained in first two sections of this paper

As in [BCBA], we have assumed that the query and data distribution are the same and that the relation `Q` contains no duplicate entities. A topic for future research is to evaluate the $l\alpha\beta$-model without these assumptions. Another topic for future research is a (formal) error analysis for the $l\alpha\beta$-model, i.e., the derivation of a (mathematical) formula that is able to predict the error caused by the model. Finally, we plan to compare the $l\alpha\beta$-model with other selectivity models, such as the parametric and non-parametric ones.

References

[ACM96] *Proceedings of the 1996 ACM SIGMOD International Conference on the Management of Data*, ACM Press, 1996.

[AOV+99] M.P. Atkinson, M.E. Orlowska, P. Valduriez, S.B. Zdonik, and M.L. Brodie (eds.), *Proceedings of the 25th VLDB Conference*, VLDB, Morgan Kaufmann, September 1999.

[BCBA] H.E. Blok, R.S. Choenni, H.M. Blanken, and P.M.G. Apers, *A selectivity model for fragmented relations in information retrieval*, IEEE Trans. on Knowledge and Data Engineering, Awaiting notification of acceptance from editor.

[BCBA01] H.E. Blok, R.S. Choenni, H.M. Blanken, and P.M.G. Apers, *A selectivity model for fragmented relations in information retrieval*, Technical Report 01-02, Centre for Telematics and Information Technology (CTIT), University of Twente, Enschede, The Netherlands, January 2001.

[BHC+01] H.E. Blok, D. Hiemstra, R.S. Choenni, F.M.G. de Jong, H.M. Blanken, and P.M.G. Apers, *Predicting the cost-quality trade-off for information retrieval queries: Facilitating database design and query optimization*, Tenth

International Conference on Information and Knowlegde Management (ACM CIKM'01), ACM SIGIR/SIGMIS, ACM Press, November 2001.

[Blo02] H.E. Blok, *Database Optimization Aspects for Information Retrieval*, Ph.D. thesis, University of Twente, Enschede, The Netherlands, April 2002, ISBN 903651732X.

[BVBA01] H.E. Blok, A.P. de Vries, H.M. Blanken, and P.M.G. Apers, *Experiences with IR TOP N Optimization in a Main Memory DBMS: Applying 'the Database Approach' in New Domains*, Advances in Databases, 18th British National Conference on Databases (BNCOD 18) (Chilton, UK) (B. Read, ed.), Lecture Notes in Computer Science, vol. 2097, CLRC Rutherford Appleton Laboratory, Springer, July 2001.

[Car75] A.F. Cardenas, *Analysis and performance of inverted data base structures*, Communications of the ACM **18** (1975), no. 5, 253–263.

[CG99] S. Chaudhuri and L. Gravano, *Evaluating Top-k Selection Queries*, In Atkinson et al. [AOV+99], pp. 397–410.

[CK98] M.J. Carey and D. Kossmann, *Reducing the Braking Distance of an SQL Query Engine*, In Gupta et al. [GSW98], pp. 158–169.

[CKSA96] R.S. Choenni, M.L. Kersten, A. Saad, and J. van den Akker, *A Framework for Multi-Query Optimization*, Report CS-R9638, CWI, Centre for Mathematics and Computer Science, Amsterdam, The Netherlands, 1996.

[CMN98] S. Chaudhuri, R. Motwani, and V. Narasayya, *Random Sampling for Histogram Construction: How much is enough?*, Proceedings of the 1998 ACM SIGMOD International Conference on the Management of Data, ACM Press, 1998, pp. 436–447.

[CMN99] S. Chaudhuri, R. Motwani, and V. Narasayya, *On Random Sampling over Joins*, Proceedings of the 1999 ACM SIGMOD International Conference on the Management of Data, ACM Press, 1999, pp. 263–274.

[CR94] C.M. Chen and N. Roussopoulos, *Adaptive Selectivity Estimation Using Query Feedback*, Proceedings of the 1994 ACM SIGMOD International Conference on the Management of Data, ACM Press, May 1994, pp. 161–172.

[DR99] D. Donjerkovic and R. Ramakrishnan, *Probabilistic Optimization of Top N Queries*, In Atkinson et al. [AOV+99], pp. 411–422.

[FSGM+98] M. Fang, N. Shivakumar, H. Garcia-Molina, R. Motwani, and J.D. Ullman, *Computing Iceberg Queries Efficiently*, In Gupta et al. [GSW98], pp. 299–310.

[GGMS96] S. Ganguly, P.B. Gibbons, Y. Matias, and A. Silberschatz, *Bifocal Sampling for Skew-Resistant Join Size Estimation*, In *Proceedings of the 1996 ACM SIGMOD International Conference on the Management of Data* [ACM96], pp. 271–281.

[GSW98] A. Gupta, O. Shmueli, and J. Widom (eds.), *Proceedings of the 24th VLDB Conference*, VLDB, Morgan Kaufmann, 1998.

[HATB98] J. Hair, R. Anderson, R. Tatham, and W. Black, *Mutlivariate Data Analysis*, 5th ed., Prentice Hall, Inc, New Jersey, 1998, ISBN 0-13-930587-4.

[HKWY97] L.M. Haas, D. Kossmann, E.L. Wimmers, and J. Yang, *Optimizing Queries across Diverse Data Sources*, Proceedings of the 23th VLDB Conference, VLDB, 1997.

[IB86] A. IJbema and H.M. Blanken, *Estimating bucket accesses: A practical approach*, Proceedings of the Second International Conference on Data Engineering (ICDE'86), IEEE Computer Society, IEEE, February 1986, pp. 30–37.

[IP95] Y.E. Ioannidis and V. Poosala, *Balancing Histogram Optimality and Practicality for Query Result Size Estimation*, Proceedings of the 1995 ACM SIGMOD International Conference on the Management of Data, ACM Press, 1995, pp. 233–244.

[LNS90] R.J. Lipton, J.F. Naughton, and D.A. Schneider, *Practical Selectivity Estimation through Adaptive Sampling*, Proceedings of the 1990 ACM SIGMOD International Conference on the Management of Data (H. Garcia-Molina and H.V. Jagadish, eds.), ACM Press, June 1990, pp. 1–11.

[PIHS96] V. Poosala, Y.E. Ioannidis, P.J. Haas, and E.J. Shekita, *Improved Histograms for Selectivity Estimation of Range Predicates*, In *Proceedings of the 1996 ACM SIGMOD International Conference on the Management of Data* [ACM96], pp. 294–305.

[TRE] *Text REtrieval Conference (TREC)*, URL: http://trec.nist.gov/.

[Yao77] S.B. Yao, *Approximating Block Accesses in Database Organizations*, Communications of the ACM **20** (1977), no. 4, 260–261.

[Zip49] G.K. Zipf, *Human Behavior and the Principle of Least Effort*, Addison-Wesley, Reading, MA, USA, 1949.