Concise Papers

A Selectivity Model for Fragmented Relations: Applied in Information Retrieval

Henk Ernst Blok, Sunil Choenni, Henk M. Blanken,
and Peter M.G. Apers, *Member, IEEE*

Abstract—New application domains cause today's database sizes to grow rapidly, posing great demands on technology. Data fragmentation facilitates techniques (like distribution, parallelization, and main-memory computing) meeting these demands. Also, fragmentation might help to improve efficient processing of query types such as top N . Database design and query optimization require a good notion of the costs resulting from a certain fragmentation. Our mathematically derived selectivity model facilitates this. Once its two parameters have been computed based on the fragmentation, after each (though usually infrequent) update, our model can forget the data distribution, resulting in fast and quite good selectivity estimation. We show experimental verification for Zipfian distributed IR databases.

Index Terms—Selectivity, fragmentation, Zipf, information retrieval, 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 focused 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 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 are uniformly distributed. In the field of text retrieval systems, Zipf distribution of data [25] is the norm. However, for emerging advanced applications, many assumptions should be reconsidered.

In this paper, we propose a comprehensive selectivity model and derive a mathematically closed formula to predict the selectivity of a query. We assume that the query and data distributions are known a priori. In the field of information retrieval, it is assumed that the query terms and data have the same distribution, thus fulfilling our assumption since the data distribution is typically known.

Although our discussion is in the context of modeling information retrieval as a database application, the selectivity model is applicable for a number of database applications (see Section 2), such as distributed databases, autonomous databases [22], etc.

In many retrieval systems, terms are extracted from documents and indexed. This can be regarded as a binary relation

$\text{COLL}(\text{term}, \text{doc})$ in which *term* is a term and *doc* a document that is indexed by *term*. In information retrieval systems, a user expresses his query by enumerating a set of terms. Therefore, a query can be represented as a unary relation $Q(\text{term})$.

Our selectivity model is focused toward the prediction of the number of hits in relation COLL , or a fragment of it, given a set of terms. In other words, our model predicts the size of the (semi)join between the relations COLL , or a fragment of it, and Q .

To evaluate our model, we have performed a number of experiments to determine whether the predicted selectivity values obtained by our model meet the experimental selectivity values. The experimental values have been obtained by measuring the selectivity for a number of representative document collections (which are part of the, in the IR field, well-known TREC [23] data set). It appears that the predicted selectivity values closely match the measured values.

In the literature, a large number of efforts have been reported on the prediction of selectivity factors in different contexts and under different assumptions [6], [24], [18], [20], [12], [19], [15], [21], [10], [11]. Roughly, two directions can be distinguished in the prediction of selectivity factors. Research in the first direction has been focused on 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 [6] to closed mathematical formulae [24] 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: nonparametric, parametric, curve fitting, and sampling. We refer to [12] for a more detailed description of each of these.

Our problem definition fits in the second research direction. However, our approach differs on two fundamental points compared to the above-mentioned categories. First, we focus on the estimation of the selectivity for a fragmented database. In our case, the fragment size can be regarded as a parameter. Such an option 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 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. Our model is not a sampling, curve fitting, or nonparametric method. We propose a model that relies on two parameters that are computed from the data distribution. However, we do not approximate the data distribution by a "standard" distribution function. Instead, we use the real discrete distribution of the data and compute a mathematical approximation to the real estimated value for the selectivity (see Section 3 for the details). Therefore, at best, our approach can be regarded as a combination of a nonparametric and a parametric approach. To our best knowledge, such an integrated approach has not been reported before in the literature.

Finally, we want to point out that a notion of the costs related to fragmentation might be of use in top- N query optimization [8], [14], [7], [13], multiquery optimization [9], and distributed database query optimization [17]. The relationship between our research and these topics becomes more clear in Section 2.

The remainder of this paper is structured as follows: First, we discuss our problem in more detail. Then, we derive our model in Section 3. In Section 4, we present experimental verification of our model. Finally, Section 5 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,

• H.E. Blok, H.M. Blanken, and P.M.G. Apers are with the Computer Science Department, University of Twente, PO Box 217, 7500 AE, Enschede, The Netherlands.

E-mail: {h.e.blok, h.m.blanken, p.m.g.apers}@utwente.nl.

• S. Choenni is with the Computer Science Department, University of Twente, PO Box 217 7500 AE, Enschede, The Netherlands and the University Nyenrode, Straatweg 25, 3621 BG, Breukelen, The Netherlands. E-mail: s.choenni@nyenrode.nl.

Manuscript received 27 Apr. 2001; revised 17 Dec. 2001; accepted 1 Aug. 2002.

For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 114058.

decision support systems, and multimedia databases. To meet the performance demands of these applications, a widely used strategy is to exploit main-memory capacity by loading a 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*.

In defining our selectivity model, we take into account the aspect of partitioning. As noted before, our discussion is in the context of information retrieval. Let us consider the following three key relations:

Term-document pairs (1). Each time a term occurs in a certain document, *doc*, a *term-doc* pair is recorded in the relation *COLL*. As mentioned before, this relation, which actually is an *inverted list*, is grouped by term and then ordered on ascending group count.¹ This relation is usually Zipfian distributed and very large.

$$COLL(term, doc) \quad (1)$$

Document frequencies (2). The *DocFreq* relation contains, for every term, its document frequency, *df*. The *df* of a term is the number of documents that term occurs in and equals the group count of the term in *COLL*. We assume that *DocFreq* is ordered ascending on *df*, meaning that it is ordered similar to *COLL*.

$$DocFreq(term, df) \quad (2)$$

Query (3). This relation is nothing more than a set of terms, constructed each time a user query is presented to the system for evaluation.

$$Q(term) \quad (3)$$

Let us assume that we are only interested in the first *m* tuples of relation *DocFreq* and their corresponding tuples in *COLL*, which is *COLL'*. Our problem definition is to predict the size of the (semi)join between *COLL'* and *Q*. For several reasons, such a prediction is useful [3].

3 MATHEMATICAL MODEL

In this section, we present a mathematical approach to our selectivity problem. To make it more accessible, we first introduce some terminology. Having presented that, we present another problem, analogous to the original one, which demonstrates the key issues more intuitively. Using this analogy, we describe a mathematical model for the estimated selectivity ratio.

3.1 Preliminary Definitions

We introduce a mathematical notation corresponding to the relational definitions presented in Section 2. First of all, we identify terms (4) and we assume that our vocabulary consists of *n* *unique* terms (5). For the sake of convenience, *T* is ordered such that *t*₁ is the most discriminative term (i.e., it occurs in the least number of documents of all terms) and *t*_{*n*} the least discriminative term (i.e., it occurs in the highest number of documents of all

terms). Second, we identify documents (6) and the set of all documents (7) known to our system. A document is defined as a list of terms. In the following, we mean by *t*_{*i*} ∈ *T* the *i*th term in list *T*. Using these notions, we define the set of all term-document pairs known to our system (8). A pair (*t*_{*i*}, *d*_{*j*}) means that term *t*_{*i*} occurs in document *d*_{*j*}. This is the set representation of the *COLL(term, doc)* relation, as defined previously. Based on this set, we define the document frequency per term *t*_{*i*} (9) and *DF* is the list of document frequencies for all *t*_{*i*} (10). Finally, we define the set of query terms and the query length (11). Note that we assume no *out-of-vocabulary* words, meaning that all terms occurring in a query are known to the system (after stemming). For sufficiently large data collections, *T* will be so large that this assumption is reasonable. We might have to reconsider this assumption if it appears that our selectivity model does not perform well enough in practice.

$$t_i \equiv \text{"a term"}, \quad (4)$$

$$T \equiv [t_1, t_2, t_3, \dots, t_i, \dots, t_n], \quad (5)$$

$$d_j \equiv \text{"a document"} = [t_i | t_i \text{ is "a term"}], \quad (6)$$

$$D \equiv \{d_1, d_2, d_3, \dots, d_j, \dots, d_{|D|}\}, \quad (7)$$

$$COLL \equiv \{(t_i, d_j) | t_i \in d_j \in D\}, \quad (8)$$

$$df_i \equiv |\{(t_i, d_j) | \forall j : t_i \in d_j \in D\}|, \quad (9)$$

$$DF \equiv [df_1, df_2, df_3, \dots, df_i, \dots, df_n], \quad (10)$$

$$Q \equiv \{t_i | t_i \in T\}, |Q| = l, \quad (11)$$

$$T' \equiv [t_1, t_2, t_3, \dots, t_i, \dots, t_m], \quad \text{where } m \leq n, \quad (12)$$

$$Q' \equiv \{t_i | t_i \in Q \wedge t_i \in T'\}, \quad (13)$$

$$COLL' \equiv \{(t_i, d_j) | t_i \in T' \wedge t_i \in d_j \in D\} \subset COLL, \quad (14)$$

$$COLL'_Q \equiv \{(t_i, d_j) | t_i \in Q' \wedge t_i \in d_j \in D\} \subset COLL', \quad (15)$$

$$DF' \equiv [df_1, df_2, df_3, \dots, df_m], \quad \text{where } m \leq n. \quad (16)$$

As explained before, we are only interested in the most discriminative terms, say *m* of the *n* terms in total. Since we assumed *T* to be ordered from most to least discriminative, we can take the *m* first elements, resulting in *T'* (12). The other expressions (13)-(16) follow directly from this and their corresponding versions (4)-(11).

We can now define the selectivity ratio as the relative fraction of the most discriminative collection part selected by the query. It is this fraction we want to estimate:

$$f_{\sigma_{measured}} \equiv \frac{|COLL'_Q|}{|COLL'|}. \quad (17)$$

3.2 An Analogy: Playing Math Darts

To make the discussion more accessible, we reformulate our problem in terms of a dart game.

As an analogy for our *COLL* relation, we construct a special dart board, as shown in Fig. 1. The board consists of concentric rings (so no sectors like one is accustomed to for the normal dart

1. In reality, this relation usually contains more attributes like the term frequency (*tf*). Since these extra attributes are not relevant to the remainder of this paper, we have left them out for clarity.

Fig. 1. Our mathematical dartboard: Each ring i has surface area df_i .

board). The surface area of each ring i is equal to df_i , where the “bulls eye” in our case corresponds to the records in COLL belonging to term 1 and the outer most ring to term n .

A query of length l can be seen as throwing l darts completely aselectively at our dartboard (“just” tell the darts player to do so instead of the usual nonaselect aiming for the bull). Note that this means that the probability of hitting a ring is therefore proportional to its surface area.

Also, we do not allow one ring to be hit more than once. In case a dart hits a ring that has been hit already, the dart has to be taken out and thrown again (corresponding to the property that each query term occurs only once in a query). Furthermore, we assume that our dartboard is so huge that each dart hits the board (corresponding to our “no out-of-vocabulary terms” assumption stated before).

After l darts have been thrown at the board, we compute the total surface area of the rings hit with ordinal number less than or equal to m , i.e., add the surface area of each gray ring in Fig. 1 that has been hit to the total.

The problem we want to solve now is rephrased as: *What is the expected total surface area of the rings hit within the inner m rings?* We use this problem in our modeling efforts, instead of referring to the original selectivity problem.

3.3 Selectivity Model

In this section, we first introduce some additional definitions. Next, we use these definitions to finally construct a mathematical expression describing the desired expected surface area of rings hit within the inner m rings.

3.3.1 Additional Mathematical Definitions

In (18)-(23), we introduce some additional mathematical definitions to formalize our problem. Furthermore, we list some utility functions in (24)-(26) for supporting purposes. For an elaborated description of these definitions and functions, we refer to [3].

$(X = x) \equiv$ The total surface area of the inner m rings that are hit equals x , given that l darts have been thrown at the board.

(18)

$(K = k) \equiv$ The event that k of the given l darts hits the inner m rings, $k \leq l$.

(19)

$p_i \equiv \frac{df_i}{|COLL'|} =$ The probability of a dart hitting ring i of the inner m rings.

(20)

$\vec{p}_i = (p_1, p_2, p_3, \dots, p_m)$.

(21)

$\alpha \equiv \sum_{i=1}^m \frac{df_i^2}{\left(\sum_{i=1}^m df_i\right)^2} = \frac{\sum_{i=1}^m df_i^2}{|COLL'|^2}$.

(22)

$\beta \equiv \sum_{i=1}^m \frac{df_i}{\sum_{i=1}^m df_i} = \frac{|COLL'|}{|COLL|}$.

(23)

$SDF_Z \equiv \sum_{i \in Z} df_i$, where $Z \subset \{1, 2, 3, \dots, m\}$.

(24)

$h_k : \vec{p} \mapsto \sum_{Z \subset \{1, 2, 3, \dots, m\} \atop |Z|=k} \prod_{i \in Z} p_i$.

(25)

$g_k : \vec{p} \mapsto \left(\sum_{i=1}^m p_i\right)^k$.

(26)

Using the definitions (18)-(26), we now can define the probability that k of the l darts (i.e., query terms) hit the inner m ring area (27). Furthermore, we can define the conditional probability that the surface area of rings hit in the inner m ring area is x , given that k darts hit that area (28). The conditional expected surface area of hit rings in the inner m ring area (29) follows directly from (28). In turn, the unconditional version of this expected value (30) follows directly from (29) and (27). Actually, we are interested in the relative surface area hit in the inner m ring area (i.e., the selectivity ratio). This means that we have to divide the value of (30) by the total surface area of the inner m ring area $|COLL'|$ to get the definition of the estimated selectivity ratio (31).

$$P(K = k) = \binom{l}{k} \beta^k (1 - \beta)^{(l-k)}, \quad (27)$$

$$P(X = x | K = k) = \sum_{Z \subset \{1, 2, 3, \dots, m\} \atop SDF_Z = x \atop |Z|=k} \prod_{i \in Z} p_i, \quad (28)$$

$$E(X | K = k) \equiv \sum_x P(X = x | K = k) x, \quad (29)$$

$$E(X) = \sum_{k=1}^l E(X | K = k) P(K = k), \quad (30)$$

$$f_{\sigma_{\text{theoretical}}} \equiv \frac{E(X)}{|COLL'|}. \quad (31)$$

TABLE 1
TREC Document Collection Statistics (after Stemming and Stopping)

Collection	# terms	# term-document pairs	# documents
FT	175,593	26,544,084	210,158
LATIMES	176,853	19,565,029	131,890
CR	69,433	5,189,218	27,921

3.3.2 Actual Model Construction

Now, we are able to derive our final model, mainly by means of manipulating the given formulas. We start by using (28) to expand (29) and rewrite it into a more useful form:

$$E(X|K=k) = \sum_{\substack{Z \subset \{1,2,3,\dots,m\} \\ |Z|=k}} \left(\prod_{i \in Z} p_i \right) \left(\sum_{i \in Z} df_i \right). \quad (32)$$

If we take (32) and use it to expand (30), we get:

$$E(X) = \sum_{k=1}^l \sum_{\substack{Z \subset \{1,2,3,\dots,m\} \\ |Z|=k}} \left(\prod_{i \in Z} p_i \right) \left(\sum_{i \in Z} df_i \right) P(K=k). \quad (33)$$

Via regrouping of the terms in the inner sums, we get:

$$E(X) = \sum_{k=1}^l \sum_{i'=1}^m p_{i'} df_{i'} \sum_{\substack{Z \subset \{1,2,3,\dots,m\} \setminus i' \\ |Z|=k-1}} \prod_{i \in Z} p_i P(K=k). \quad (34)$$

The " $\prod_{i \in Z} p_i$ "-part can be replaced by an expression in g_k , using (25) and (26):

$$\begin{aligned} E(X) &= \sum_{k=1}^l \sum_{i'=1}^m p_{i'} df_{i'} \frac{\partial}{\partial p_{i'}} (h_k(\vec{p})) P(K=k) \\ &\leq \sum_{i'=1}^m p_{i'} df_{i'} \sum_{k=1}^l k g_{k-1}(\vec{p}) P(K=k). \end{aligned} \quad (35)$$

From the definitions of g_k and p_i ((26) and (20), respectively) follows directly:

$$E(X) \leq \sum_{i'=1}^m p_{i'} df_{i'} \sum_{k=1}^l k 1^{k-1} \binom{l}{k} \beta^k (1-\beta)^{(l-k)} \leq \sum_{i'=1}^m p_{i'} df_{i'} l \beta. \quad (36)$$

If we substitute $E(X)$ and (22) in (31), we get:

$$f_{\sigma_{\text{theoretical}}} \equiv \frac{E(X)}{|COLL|} \leq l \alpha \beta. \quad (37)$$

We now define our estimator for the expected selectivity ratio as the right-hand side of this expression, as shown in (38):

$$f_{\sigma_{\text{estimated}}} \equiv l \alpha \beta \quad (38)$$

This completes our model.

4 EXPERIMENTAL VERIFICATION

This section describes the test databases that have been used and the experimental results that have been obtained on them.

4.1 Setup

As mentioned before, we used document collections from the TREC set for the evaluation of our selectivity model (see Table 1).

We used the 50 retrieval queries, also known as topics in the IR field, from TREC-6. These queries range in length (l) from 9 up to and including 61 terms with an average of 27.441 terms. For all three collections, we evaluated our selectivity model for 11 cases of m , with m such that

$$\frac{m}{n} \in \{0.9, 0.925, 0.94, 0.95, 0.96, 0.97, 0.98, 0.985, 0.99, 0.995, 0.999\}.$$

The main reason for selecting the fraction of used terms from the interval $[0.9, 0.999]$ is that these situations concern the most critical area of the Zipf distribution of the data. Below 0.9, the distribution is so flat that it is almost uniform and, therefore, not really interesting. We excluded fractions above 0.999 from our results since the only logical next fraction would have been 1.0. However, that case is handled quite differently by our experimental DBMS since it then has no fragmentation at all, resulting in some practical problems to compare the outcome with the rest. Next to that, the case for 1.0 did not seem to have any additional impact on the overall conclusions of our model compared to the case for 0.999.

4.2 Results

We ran the 50 queries for each of the 11 choices of m (relative to n), resulting in 550 cases per collection. For each of these cases, we computed the exact fraction of retrieved tuples in $COLL'$, using (17), and the estimated fraction, using (38). In Fig. 2a, we plotted, for each of the 550 cases for the FT collection, the points ($f_{\sigma_{\text{measured}}}, f_{\sigma_{\text{estimated}}}$). We also plotted the line where the points should be located ideally, in case our model is perfect. In a similar manner, Fig. 2b and Fig. 2c show the results for the LATIMES and CR collection, respectively. As one can clearly see, for all three collections, the point clouds are nicely arranged along the ideal line, demonstrating that our model indeed performs quite well.

5 CONCLUSIONS AND FUTURE RESEARCH

Our model is useful in advanced application domains such as data warehousing and information retrieval. In those domains, the amount of data that needs to be stored becomes huge, requiring techniques like distribution, partitioning, and parallelizing. Also, main-memory computing is often regarded as a technique that can help to improve the efficiency of resource utilization in those new domains. All these techniques can be facilitated by fragmentation. A tool needed to allow exploiting fragmentation in an effective way is a notion of the costs related to it. Our selectivity model is an attempt to meet this need.

Our model only uses two parameters, α and β , which need to be determined before the model can be used. However, these parameters can easily be updated incrementally. Furthermore, we demonstrated the accuracy of our model by testing it for three information retrieval databases. The results show that the estimated selectivity values match the real selectivity value, measured during these tests, quite well.

Some topics for future research can be distinguished as well. First of all, a mathematical analysis of the error of our model is needed to allow better comparison with other methods. We have already been working on such an analysis, but still have to take care of some mathematical issues to complete it. Second, we have only derived a method for two fragments in this paper. The extension of the model to work for more than two fragments can be obtained without much effort by following the same approach as we did here. Furthermore, we used the data distribution of the inverted list to represent the distribution of the terms within a query. For an information retrieval case, this is perfectly correct. The reason for this is that queries are assumed to be a representation of the documents a user wants to retrieve, thus being equally distributed in the term domain. Other applications might not have such a nice feature. Probably one needs to learn the distribution using statistical regression techniques. Another interesting research question is the effect of ignoring parts of the COLL relation on

Fig. 2. Estimated selectivity ratio versus measured selectivity ratio. (a) FT collection. (b) LATIMES collection. (c) CR collection.

the quality of the output of a query. In [4], a model is reported that predicts the quality for various subsets of COLL. In the future, we want to link the model reported in [4] to our selectivity model. An extended version of this work can be found in [3], [5].

REFERENCES

- [1] *Proc. ACM SIGMOD*, 1996.
- [2] *Proc. Very Large Data Base Conf.*, M.P. Atkinson, M.E. Orlowska, P. Valduriez, S.B. Zdonik, and M.L. Brodie, eds., 1999.
- [3] H.E. Blok, S. Choenni, H.M. Blanken, and P.M.G. Apers, *A Selectivity Model for Fragmented Relations in Information Retrieval*, Technical report, no. TR-CITIT-01-02, Centre for Telematics and Information Technology, University of Twente, The Netherlands, Jan. 2001.
- [4] H.E. Blok, D. Hiemstra, S. Choenni, F. 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," *Proc. ACM Conf. Information and Knowledge Management*, 2001.
- [5] H.E. Blok, "Database Optimization Aspects for Information Retrieval," PhD thesis, Univ. of Twente, Enschede, The Netherlands, 2002.
- [6] A.F. Cardenas, "Analysis and Performance of Inverted Data Base Structures," *Comm. ACM*, vol. 18, no. 5, pp. 253-263, 1975.
- [7] S. Chaudhuri and L. Gravano, "Evaluating Top- k Selection Queries," *Proc. Very Large Data Base Conf.*, pp. 397-410, 1999.
- [8] M.J. Carey and D. Kossmann, "Reducing the Braking Distance of an SQL Query Engine," *Proc. Very Large Data Base Conf.*, pp. 158-169, 1998.
- [9] S. Choenni, M. Kersten, A. Saad, and J. v.d. Akker, "A Framework for Multi-Query Optimization," Report CS-R9638, CWI, Centre for Math. and Computer Science, Amsterdam, 1996.
- [10] S. Chaudhuri, R. Motwani, and V. Narasayya, "Random Sampling for Histogram Construction: How Much Is Enough?" *Proc. ACM SIGMOD*, pp. 436-447, 1998.
- [11] S. Chaudhuri, R. Motwani, and V. Narasayya, "On Random Sampling over Joins," *Proc. ACM SIGMOD*, pp. 263-274, 1999.
- [12] C.M. Chen and N. Roussopoulos, "Adaptive Selectivity Estimation Using Query Feedback," *Proc. ACM SIGMOD*, pp. 161-172, 1994.
- [13] D. Donjerkovic and R. Ramakrishnan, "Probabilistic Optimization of Top N Queries," *Proc. Very Large Data Base Conf.*, pp. 411-422, 1999.
- [14] M. Fang, N. Shivakumar, H. Garcia-Molina, R. Motwani, and J.D. Ullman, "Computing Iceberg Queries Efficiently," *Proc. Very Large Data Base Conf.*, pp. 299-310, 1998.
- [15] S. Ganguly, P.B. Gibbons, Y. Matias, and A. Silberschatz, "Bifocal Sampling for Skew-Resistant Join Size Estimation," *Proc. ACM SIGMOD*, pp. 271-281, 1996.
- [16] *Proc. Very Large Data Base Conf.*, A. Gupta, O. Shmueli, and J. Widom, eds., 1998.
- [17] L.M. Haas, D. Kossmann, E.L. Wimmers, and J. Yang, "Optimizing Queries across Diverse Data Sources," *Proc. Very Large Data Base Conf.*, 1997.
- [18] A. Ijbema and H.M. Blanken, "Estimating Bucket Accesses: A Practical Approach," *Proc. Int'l Conf. Data Eng.*, pp. 30-37, 1986.
- [19] Y.E. Ioannidis and V. Poosala, "Balancing Histogram Optimality and Practicality for Query Result Size Estimation," *Proc. ACM SIGMOD*, pp. 233-244, 1995.
- [20] R.J. Lipton, J.F. Naughton, and D.A. Schneider, "Practical Selectivity Estimation through Adaptive Sampling," *Proc. ACM SIGMOD*, H. Garcia-Molina and H.V. Jagadish, eds., pp. 1-11, 1990.
- [21] V. Poosala, Y.E. Ioannidis, P.J. Haas, and E.J. Shekita, "Improved Histograms for Selectivity Estimation of Range Predicates," *Proc. ACM SIGMOD*, pp. 294-305, 1996.
- [22] A.P. Sheth and J.A. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," *ACM Computing Surveys*, vol. 22, no. 3, pp. 183-236, 1990.
- [23] A.P. de Vries and D. Hiemstra, "The miRRor DBMS at TREC," *Proc. Text Retrieval Conf.*, E.M. Voorhees and D.K. Harman, eds., pp. 725-734, NIST special publications, 1999.
- [24] S.B. Yao, "Approximating Block Accesses in Database Organizations," *Comm. ACM*, vol. 20, no. 4, pp. 260-261, 1977.
- [25] G.K. Zipf, *Human Behavior and the Principle of Least Effort*. Reading, Mass.: Addison-Wesley, Reading, 1949.