Bull. Math. Soc. Sci. Math. Roumanie
Tome 57(105) No. 3, 2014, 319–325

Some estimations in database queries

by
Silviu-Laurenţiu Vasile

Abstract

In the frame of the previous research it is asummed that relations tuples are identically distributed and the values of different attributes are independent. These assumptions are not realistic because attributes values can be imprecise and there can be a dependency between attributes. In this paper we extend the concept of selectivity factor, associated to a query to the notion of selectivity factor associated to queries sets and we give some practical results in the case of random databases.

Key Words: Database, selectivity factor, relational operators, cardinality.
2010 Mathematics Subject Classification: Primary 68P15, Secondary 68P20, 68P10.

1 Introduction

In this paper we propose to extend the concept of selectivity factor associated to a query to the notion of selectivity factor associated to a queries set $\{C_1, \ldots, C_n\}$. The framework in which we place this concept assumes that the tables of the database are updated dynamically. So, at distinct moments of time, the selectivity factor associated to each query is different. Suppose that the query C_i has the selectivity factor $p_i = \frac{\alpha_i}{k}$, where k is the number (constant) of lines of the table, and α_i is the number of lines selected by the query C_i. In this case, the selectivity factor of the queries set can be estimated by the variable:

$$\overline{p} = \frac{1}{n}\sum_i p_i = \frac{1}{n}\sum_i \frac{\alpha_i}{k} = \frac{\sum\limits_i \alpha_i}{nk}. \tag{1.1}$$

The mean of $\overline{p}$ is given by $E(\overline{p}) = p$ and it represents the (theoretical) selectivity factor of the queries set, if p is an unbiased estimation for $\overline{p}$. The variance of this variable is $Var(\overline{p}) = \dfrac{p(1-p)}{n} \leq \dfrac{1}{4n}$, and the standard deviation is $\sigma \leq \dfrac{1}{2\sqrt{n}}$.

From the Chebyshev's inequality, we get:

$$P(|\overline{p} - p| < t\sigma) \geq 1 - \frac{1}{t^2}. \tag{1.2}$$

We denote $1-\frac{1}{t^2}=1-\delta$, and from here it results that $t_\delta=\frac{1}{\sqrt{\delta}}$. We denote $t\sigma=\epsilon\cong 0,01$. For a small value of ϵ we obtain a large value $1-\delta$, so δ is small.

From the preceding notations, it implies that:

$$t_\delta\cdot\frac{1}{2\sqrt{n}}\le 0,01. \tag{1.3}$$

In order to be able to estimate the selectivity factor the number n of queries must satisfy the condition $n\ge\frac{t_\delta^2}{4\cdot(0,01)^2}$.

2 Estimation of the selectivity factor

Assuming that the tables are updated dynamically, we consider that the number of lines of the tables at the moments of different queries is k_i, uniformily distributed. Let's assume that the minimum, respectively, maximum values of this variable are known: $m\le k_i\le n$. Then, by means of the simulation methods of the uniform variable, we obtain an estimation of the generalized selectivity factor for a queries set through the following algorithm:

Algorithm AEGSF (Algorithm for the estimation of the generalized selectivity factor)

```
Input: $t_\delta$.
Step 1. Determine $N_0=[\frac{t_\delta^2}{4\cdot(0,01)^2}]+1$. Choose $N\ge N_0$.
Step 2. For $i=1$ to $N$ do
    Begin
        Generate a random value $U\in(0,1)$.
        Determine $k_i=m+[(n-m)U]+1$.
        Calculate $p_i=\frac{\alpha_i}{k_i}$.
    End.
Step 3. Determine $\overline{p}=\frac{1}{N}\sum_i p_i$.
```

Given values $p_1,p_2,\ldots,p_N$ generated at step 2, the selectivity factor of the queries set can be estimated by $\overline{p}$.

In many cases, in the frame of the statistic profiles used in the previous research, it was assummed that the tuples of the relations are identically distributed, relative to the values of an attribute, and the values of different attributes are independent.

This assumption is not realistic, there ar many examples in databases where attributes that exhibit dependencies between and/or their values ar imprecise[7]. Due to this reason, there have been developed several methods to estimate the selectivity factor. Among them the most common are: parametric and non-parametric methods, and the maximum entropy principle [1].

A subsequent classification of the methods for estimation of the selectivity factor used traditionally, include the following estimator types [4]:

- methods based on selection, which determine the selectivity factor only on the basis of the information at runtime, without using the information collected previously;
- parametric methods, which use only the information collected previously, ignoring the on-line information.

The disadvantage of the first class of estimators consists in the insufficient use of the available information, whereas the second class leads to an unprecise estimation in an environment with frequent updates. There have been proposed hybrid estimators[4], which weight the two types enumerated above and whose results have been validated in practice.

3 Hybrid estimator for the generalized selectivity factor

In the following we present a hybrid estimator for the selectivity factor of a query and we will extend the results for the case of the selectivity factor associated to a queries set, introduced before.

In the case of a single query, we consider f the characteristic function of a selection predicate, and x_i a tuple. The function f can be defined in the following way:

$$y_i = f(x_i) = \begin{cases} 1, & \text{if } x_i \text{ satisfies the selection predicate;} \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

In the approach based on selection, the main technique consists in choosing randomly, repeatedly, a tuple in the table on the basis of the queries' predicate, followed by the inference about the real selectivity, using the estimated selectivity obtained from the sample data. Thus, one can realize the inference and according by obtain an approximation for the real selectivity p as:

$$\hat{p}_n = \frac{1}{n} \cdot \sum_{i=1}^{n} y_{r_i} = \frac{1}{n} \cdot \sum_{i=1}^{n} f(x_{r_i}), \tag{3.2}$$

where n represents the sample size, k is the total number of tuples and the index r_i is a random integer, between 1 and k. Consequently, the average total number of tuples which satisfy the selection is $k \cdot \hat{p}_n$.

A hybrid estimator $\breve{p}_n$ of the selectivity factor is given by a linear combination between the estimated selectivity $\hat{p}_n$ and the estimated selectivity $\tilde{p}$, obtained by a parametric estimator or by a table based estimator:

$$\breve{p}_n = t \cdot \hat{p}_n + (1 - t) \cdot \tilde{p}, \tag{3.3}$$

where t is a parameter in the interval $[0, 1]$.

In order to validate an estimator, the mean-squared error (mse) is used to quantify the estimators performances:

$$mse(\overline{p}) = E(\overline{p} - p)^2 = \frac{1}{n} \cdot \sum_{i=1}^{n} (\overline{p}_i - p)^2, \tag{3.4}$$

where $\overline{p}_i$ is the individual selectivity estimated by an estimator, p is the total, real selectivity, depending on the given query, and n is the sample size.
The value mse of an estimator represents the accuracy of its estimation, as well as its safety. The smaller the mse value, the betters the estimator is. For a method based on selection, the mse value is $\dfrac{p \cdot (1-p)}{n}$. For an estimator which uses a parametric or table based method, the mse value is $(\tilde{p}-p)^2$, and $\tilde{p}$ remains unchanged for the given query, until the parametric or table based estimator is recomputed, using the updated information.
The different values of the parameter t represents different weights of the two estimators. In the extreme cases $t = 1$ or $t = 0$, the hybrid model reduces to a selection based estimator, respectively to a parametric or a table based one. The existence of an optimum value of the parameter t and the calculation of this value have been determined in the following theorem[4].

Theorem 1. *The optimum value of t, denoted by t_n^*, is given by the formula:*

$$t_n^* = \frac{(\tilde{p}-p)^2}{p \cdot \dfrac{1-p}{n} + (\tilde{p}-p)^2}. \tag{3.5}$$

The mse value for the hybrid estimator corresponding to the optimum parameter t_n^* is smaller than each of the two estimators when $0 < p < 1$ and $p \neq \tilde{p}$, meaning that:

$$E(\breve{p}_n^* - p)^2 < min\{\frac{p \cdot (1-p)}{n}, (\tilde{p}-p)^2\}, \tag{3.6}$$

where $\breve{p}_n^*$ is $\tilde{p}_n$ for $t = t_n^*$.

We can apply the hybrid estimators in the case proposed previously, of several queries, therefore of the generalized selectivity factor. We know that $p_1, ...p_q$ are the selectivity factors associated to the q queries. Consider f_i the characteristic functions of the predicates associated to each of the q queries and be x_{ij} a tuple in the query i. We consider a selection associated to each query.
The functions f_i are given by the formula:

$$y_{ij} = f_i(x_{ij}) = \begin{cases} 1, & \text{if } x_{ij} \text{ satisfies the selection predicate of the query i;} \\ 0, & \text{otherwise.} \end{cases} \tag{3.7}$$

Let n_i be the sizes of the samples associated to the q queries. Then, an approximation of the selectivity of the query i is given by:

$$\hat{p}_{n_i} = \frac{1}{n_i} \cdot \sum_{j=1}^{n_i} y_{ir_j} = \frac{1}{n_i} \cdot \sum_{j=1}^{n_i} f_i(x_{ir_j}), \tag{3.8}$$

where r_{ij} is a random integer between 1 and n_i.
The approximate number of tuples of the result of the query i will be $k_i \cdot \hat{p}_{n_i}$, where k_i is the total number of tuples of the relation in the query i.
Every query has an associated hybrid estimator:

$$\breve{p}_{n_i} = t_i \cdot \hat{p}_{n_i} + (1-t_i) \cdot \tilde{p}_i. \tag{3.9}$$

For the queries set, we obtain:

$$\overline{p} = \frac{1}{N} \cdot \sum_i \breve{p}_{n_i}, \tag{3.10}$$

where N was determined previously.
Thus, the algorithm AEGSF becomes:

Algorithm AEGSFHE (Algorithm for the estimation of the generalized selectivity, using hybrid estimators)

```
Input: $t_\delta$.
Step 1. Determine $N_0 = [\frac{t_\delta^2}{4 \cdot (0,01)^2}] + 1$. Choose $N \geq N_0$.
Step 2. For $i = 1$ to $N$ do
        Begin
            Generate a random value $U \in (0,1)$.
            Determine $n_i = m + [(n-m)U] + 1$.
            Calculate $\breve{p}_{n_i} = t_i \cdot \hat{p}_{n_i} + (1 - t_i) \cdot \tilde{p}_i$.
        End.
Step 3. Determine $\overline{p} = \frac{1}{N} \sum_i \breve{p}_{n_i}$.
```

On the basis of the values $\breve{p}_{n_1}, \breve{p}_{n_2}, \ldots, \breve{p}_{n_N}$, generated at step 2, the selectivity factor of the queries set can be estimated by $\overline{p}$, computed in the last step of the algorithm.

4 Optimization on random databases query

We have extended the notion of random database, in which the records are random vectors following a certain probability distribution, to heterogeneous random databases, in which each column can have its own unidimensional distribution.
The operators of the relational model define the operations that can be performed on the relations, in order to manipulate the database. The relational model includes the relational algebra, whose operators are either the traditional set operators (*union, intersect, product, difference*) or special relational operators (*project, select, join, division*).
The *join* operator allows the information retrieval from more correlated relations. This is a binary operation whose result is a new relation in which each tuple is a combination of a tuple in the first relation and a tuple in the second one. The required condition in order to apply the join operator is that the tuples are similar.
In case of random databases, we have the following definition of the *join* operator:

Definition 1. *Consider two relations R and S. The ϵ-join can be described as follows:*

$$join_\epsilon(R, S) = \{(x, y) \in R \times S | d(x_R, y_S) \leq \epsilon\}. \tag{4.1}$$

We denote by $N_\epsilon(join_\epsilon(R, S))$ the number of lines in the result.
During our studies[6] we have noticed the existence of some relations between the cardinalities

N=10000		**N=20000**	
Join order	**Time**	**Join order**	**Time**
$T_1 \bowtie_U T_2 \bowtie_E T_3$	39,738	$T_1 \bowtie_U T_2 \bowtie_E T_3$	256,79
$T_1 \bowtie_N T_2 \bowtie_U T_3$	37,176	$T_1 \bowtie_N T_2 \bowtie_U T_3$	229,03
$T_1 \bowtie_N T_2 \bowtie_E T_3$	33,351	$T_1 \bowtie_N T_2 \bowtie_E T_3$	199,48

Table 1: ϵ-Join operation time consuming

of the approximate join operation in different cases of the table columns' types of probability distributions. These remarks are shown in the Table 1.

We note by $T_1 \bowtie_U T_2$ the $\epsilon - join$ operation which uses columns having values distributed uniformly and with N the number of lines from each table.
The practical experiments were conducted with tables having at least three columns with their values distributed uniformly, exponentially, normally and $\epsilon < 0.01$. The best result it is obtained when we evaluate first the *join* on columns wich follows normal distribution and then on the columns which follows exponential distribution. Such result has an important impact in random database query optimization.

5 Future Work

In [7] it is proposed a new algorithm *EGO - Efficient Global Optimization* (called *Super-EGO*) to implement the ϵ-join operations. The new algorithm prevails over the others through a new technique of subsets ordering which take part in join operations and through a parallel implementation that can run on devices with multiple processors.
The basic *EGO-join* algorithm analyzes dimensions in a sequential order from 1 to n. However, for higher dimensional cases, some of the dimensions might have more discriminative power than the others. The *Super-EGO* algorithm use data sampling techniques to measure this discriminative power to make a new order. We believe that by studying the distributions of values that these dimensions could have and by using the result presented in this paper it is possible to improve the technique of reordering the dimensions enhancing the performance of this algorithm.

References

[1] S. Christodoulakis, On the estimation and use of selectivities in database performance evaluation, *Research Report CS 89-24, Department of Computer Science, University of Waterloo, Canada* (1989)

[2] T.M. Connolly, C.E. Begg, *Database Systems: A Practical Approach to Design, Implementation and Management*, 3rd ed., Addison-Wesley, 2002.

[3] L. Lim L, M. Wang, S. Padmanabhan, J. Vitter, R. Parr, XPathLearner, An On-Line Self-Tuning Markov Histogram for XML Path Selectivity Estimation, *Proceedings of the 28th VLDB Conference* (2002), 442-453

[4] Ling, Y., Sun, W., A Hybrid Estimator for Selectivity Estimation, *IEEE Transactions on Knowledge and Data Engineering*, **no. 2, vol. 11** (1999), 338-354

[5] O. Seleznjev, B. Thalheim, Random Databases with Approximate Record Matching, *Methodol. Comput. Appl. Probab.*, Springer Verlag, 2008

[6] S. L. Vasile, L. Velcescu, Relations Between Approximate Join Cardinalities in Random Database Queries, *Scientific Bulletin of "Politehnica" University of Timisoara*, no. 4, vol. 57 (2012), 247-252

[7] D. V. Kalashnikov, Super-EGO: fast multi-dimensional similarity join, *The VLDB Jurnal*, Issue 4, Volume 22 (2013), 561-585

Received: 11.12.2013

Accepted: 03.07.2014

University of Bucharest, Faculty of Mathematics,
14 Academiei str., 70109 Bucharest, Romania
E-mail: `vsl@fmi.unibuc.ro`