Consistently Estimating the Selectivity of Conjuncts of Predicates

V. Markl[1] N. Megiddo[1] M. Kutsch[2] T.M. Tran[3] P. Haas[1] U. Srivastava[4]

[1]IBM Almaden Research Center [2]IBM Germany [3]IBM Silicon Valley Lab [4]Stanford University

{marklv, megiddo, peterh}@almaden.ibm.com, kutschm@de.ibm.com, minhtran@us.ibm.com, usriv@stanford.edu

Abstract

Cost-based query optimizers need to estimate the selectivity of conjunctive predicates when comparing alternative query execution plans. To this end, advanced optimizers use multivariate statistics (MVS) to improve information about the joint distribution of attribute values in a table. The joint distribution for all columns is almost always too large to store completely, and the resulting use of partial distribution information raises the possibility that multiple, non-equivalent selectivity estimates may be available for a given predicate. Current optimizers use ad hoc methods to ensure that selectivities are estimated in a consistent manner. These methods ignore valuable information and tend to bias the optimizer toward query plans for which the least information is available, often yielding poor results. In this paper we present a novel method for consistent selectivity estimation based on the principle of maximum entropy (ME). Our method efficiently exploits all available information and avoids the bias problem. In the absence of detailed knowledge, the ME approach reduces to standard uniformity and independence assumptions. Our implementation using a prototype version of DB2 UDB shows that ME improves the optimizer's cardinality estimates by orders of magnitude, resulting in better plan quality and significantly reduced query execution times.

Proceedings of the 31st VLDB Conference, Trondheim, Norway, 2005.

1. Introduction

Estimating the selectivity of predicates has always been a challenging task for a query optimizer in a relational database management system. A classic problem has been the lack of detailed information about the joint frequency distribution of attribute values in the table of interest. Perhaps ironically, the additional information now available to modern optimizers has in a certain sense made the selectivity-estimation problem even harder.

Specifically, consider the problem of estimating the selectivity $s_{1,2,\dots,n}$ of a *conjunctive predicate* of the form $p_1 \wedge p_2 \wedge \dots \wedge p_n$, where each p_i is a *simple predicate* (also called a *Boolean Factor,* or BF) of the form "*column op literal*". Here *column* is a column name, *op* is a relational comparison operator such as "=", ">", or "LIKE", and *literal* is a literal in the domain of the column; some examples of simple predicates are 'make = "Honda"' and 'year > 1984'. By the *selectivity* of a predicate p, we mean, as usual, the fraction of rows in the table that satisfy p.[1] In older optimizers, statistics are maintained on each individual column, so that the individual selectivities $s_1, s_2, \dots, s_n$ of $p_1, p_2, \dots, p_n$ are available. Such a query optimizer would then impose an *independence assumption* and estimate the desired selectivity as $s_{1,2,\dots,n} = s_1 * s_2 * \dots * s_n$. Such estimates ignore correlations between attribute values, and consequently can be wildly inaccurate, often underestimating the true selectivity by orders of magnitude and leading to a poor choice of query execution plan (QEP).

Ideally, to overcome the problems caused by the independence assumption, the optimizer should store the multidimensional joint frequency distribution for all of the columns in the database. In practice, the amount of

[1] Note that without loss of generality each p_i can also be a disjunction of simple predicates or any other kind of predicate (e.g., subquery, IN-list). For this work we only require that the optimizer has some way to estimate the selectivity s_i of p_i.

storage required for the full distribution is exponentially large, making this approach infeasible. Researchers therefore have proposed storage of selected *multivariate statistics* (MVS) that summarize important partial information about the joint distribution. Proposals have ranged from multidimensional histograms [PI97] on selected columns to other, simpler forms of column-group statistics [IMH+04]. Thus, for predicates $p_1, p_2, \ldots, p_n$, the optimizer typically has access to the individual selectivities $s_1, s_2, \ldots, s_n$ as well as a limited collection of joint selectivities, such as $s_{1,2}$, $s_{3,5}$, and $s_{2,3,4}$. The independence assumption is then used to "fill in the gaps" in the incomplete information, e.g., we can estimate the unknown selectivity $s_{1,2,3}$ by $s_{1,2} * s_3$.

A new and serious problem now arises, however. There may be multiple, non-equivalent ways of estimating the selectivity for a given predicate. Figure 1, for example, shows possible QEPs for a query consisting of the conjunctive predicate $p_1 \wedge p_2 \wedge p_3$. The QEP in Figure 1(a) uses an index-ANDing operation ($\wedge$) to apply $p_1 \wedge p_2$ and afterwards applies predicate p_3 by a FETCH operator, which retrieves rows from a base table according to the row identifiers returned from the index-ANDing operator.

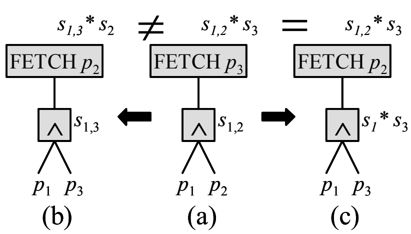

Figure 1: QEPs and Selectivity Estimation

Suppose that the optimizer knows the selectivities s_1, s_2, s_3 of the BFs p_1, p_2, p_3. Also suppose that it knows about a correlation between p_1 and p_2 via knowledge of the selectivity $s_{1,2}$ of $p_1 \wedge p_2$. Using independence, the optimizer might then estimate the selectivity of $p_1 \wedge p_2 \wedge p_3$ as $s^a_{1,2,3} = s_{1,2} * s_3$.

Figure 1(b) shows an alternative QEP that first applies $p_1 \wedge p_3$ and then applies p_2. If the optimizer also knows the selectivity $s_{1,3}$ of $p_1 \wedge p_3$, use of the independence assumption might yield a selectivity estimate $s^b_{1,2,3} = s_{1,3} * s_2$. However, this would result in an inconsistency if, as is likely, $s^a_{1,2,3} \neq s^b_{1,2,3}$. There are potentially other choices, such as $s_1 * s_2 * s_3$ or, if $s_{2,3}$ is known, $s_{1,2} * s_{2,3}/s_2$; the latter estimate amounts to a conditional independence assumption. Any choice of estimate will be arbitrary, since there is no supporting knowledge to justify ignoring a correlation or assuming conditional independence; such a choice will then arbitrarily bias the optimizer toward choosing one plan over the other. Even worse, if the optimizer does not use the same choice of estimate every time that it is required, then different plans will be costed inconsistently, leading to "apples and oranges" comparisons and unreliable plan choices

Assuming that the QEP in Figure 1(a) is the first to be evaluated, a modern optimizer would avoid the foregoing *consistency* problem by recording the fact that $s_{1,2}$ was applied and then avoiding future application of any other MVS that contain either p_1 or p_2, but not both. In our example, the selectivities for the QEP in Figure 1(c) would be used and the ones in Figure 1(b) would not. The optimizer would therefore compute the selectivity of $p_1 \wedge p_3$ to be $s_1 * s_3$ using independence, instead of using the MVS $s_{1,3}$. Thus the selectivity $s_{1,2,3}$ would be estimated in a manner consistent with Figure 1(a). Note that, when evaluating the QEP in Figure 1(a), the optimizer used the estimate $s^a_{1,2,3} = s_{1,2} * s_3$ rather than $s_1 * s_2 * s_3$, since, intuitively, the former estimate better exploits the available correlation information. In general, there may be many possible choices; the complicated (ad hoc) decision algorithm used by DB2 UDB is described in more detail in the Appendix.

Although the ad hoc method described above ensures consistency, it ignores valuable knowledge, e.g., of the correlation between p_1 and p_3. Moreover, this method complicates the logic of the optimizer, because cumbersome bookkeeping is required to keep track of how an estimate was derived initially and to ensure that it will always be computed in the same way when costing other plans. Even worse, ignoring the known correlation between p_1 and p_3 also introduces *bias* towards certain QEPs: if, as is often the case with correlation, $s_{1,3} >> s_1 * s_3$, and $s_{1,2} >> s_1 * s_2$, and if $s_{1,2}$ and $s_{1,3}$ have comparable values, then the optimizer will be biased towards the plan in Figure 1(c), even though the plan in Figure 1(a) might be cheaper, i.e., the optimizer thinks that the plan in Figure 1(c) will produce fewer rows during index-ANDing, but this might not actually be the case. In general, an optimizer will often be drawn towards those QEPs about which it knows the least, because use of the independence assumption makes these plans seem cheaper due to underestimation. We call this problem "fleeing from knowledge to ignorance".

In this paper, we provide a novel method for estimating the selectivity of a conjunctive predicate; the method exploits and combines all of the available MVS in a principled, consistent, and unbiased manner. Our technique rests on the principle of maximum entropy (ME) [GS85], which is a mathematical embodiment of Occam's Razor and provides the "simplest" possible selectivity estimate that is consistent with all of the available information. (In the absence of detailed knowledge, the ME approach reduces to standard uniformity and independence assumptions.) Our new approach avoids the problems of inconsistent QEP comparisons and the flight from knowledge to ignorance.

We emphasize that, unlike DB2's ad hoc method or the method proposed in [BC02] (which tries to choose the "best" of the available MVS for estimating a selectivity) the ME method is the first to exploit *all* of the available MVS and actually refine the optimizer's cardinality model beyond the information explicitly given by the statistics. Also, as discussed in Section 6, our results differ from virtually all current and previous work in this area, which deals only with constructing [BC04, IMH+04, BC03, SHM+05], storing [PIH+96], and maintaining [SLM+01, BCG01, AC99] multivariate statistics. Indeed our method can be used in conjunction with any of the foregoing techniques.

Thus the contributions of our paper are: (1) enunciating and formalizing the problem of consistency and bias during QEP evaluation in the presence of partial knowledge about the joint frequency distribution (as embodied in the available MVS), (2) proposing a new method for cardinality estimation in this setting that exploits all available distributional information, (3) applying the iterative scaling algorithm to compute consistent and unbiased selectivity estimates based on our problem formulation using the ME principle, (4) providing a detailed experimental evaluation of our approach with respect to quality and computation time, as well as a comparison to the DB2 UDB optimizer. Our work appears to be the first to apply information-theoretic ideas to the problem of producing consistent selectivity estimates.

The paper is organized as follows. Section 2 gives some background and formalizes the selectivity-estimation problem. In Section 3, we describe the ME approach to unbiased, efficient, and consistent selectivity estimation. We show how the iterative scaling algorithm can be applied in our setting; this well-known algorithm uses a Lagrange-multiplier approach to numerically compute an approximate ME solution. Section 4 provides an experimental evaluation, and in Section 5 we discuss some practical considerations. After surveying related work in Section 6, we conclude in Section 7. The Appendix describes the current state of the art in using MVS for cardinality estimation in a commercial DBMS.

2. Background

Commercial query optimizers [ATL+03, IBM02, IBM04, Mic04] use statistical information on the number of rows in a table and the number of distinct values in a column to compute the selectivity of a simple predicate p. Assuming ten distinct values in the MAKE column and using the *uniformity assumption*, the selectivity of the predicate p_1: 'MAKE = "Honda"' is estimated as $s_1 = 1/10$. Similarly, with 100 distinct values in the MODEL column and 10 distinct values in the COLOR column, we obtain $s_2 = 1/100$ for p_2: MODEL = "Accord" and $s_3 = 1/10$ for p_3: COLOR = "red". Advanced commercial optimizers can improve upon these basic estimates by maintaining frequency histograms on the values in individual columns.

As indicated previously, in the absence of other information, current optimizers compute the selectivity of a conjunctive predicate using the independence assumption. For instance, $p_{1,2,3} = p_1 \wedge p_2 \wedge p_3$ is the predicate restricting a query to retrieve all red Honda Accords, and the selectivity of $p_{1,2,3}$ is computed as $s_{1,2,3} = s_1 * s_2 * s_3$. In our example, the optimizer would estimate the selectivity of red Honda Accords to be 1/10000. As only Honda makes Accords, there is a strong correlation between these two columns, actually a functional dependency in this case. The actual selectivity of $p_{1,2}$ must be 1/100. Thus a more appropriate estimate of the selectivity of $p_{1,2,3}$ is 1/1000, one order of magnitude greater than the estimate using the independence assumption.

2.1 Formalizing the Selectivity Estimation Problem

We now formalize the problem of selectivity estimation for conjunctive predicates, given partial MVS, and define some useful terminology. Let $P = \{p_1, \ldots, p_n\}$ be a set of BFs. For any $X \subseteq N = \{1, \ldots, n\}$, denote by p_X the conjunctive predicate $\wedge_{i \in X} p_i$. Let s be a probability measure over 2^N, the powerset of N, with the interpretation that s_X is the selectivity of the predicate p_X. Usually, for $|X| = 1$, the histograms and column statistics from the system catalog determine s_X and are all known. For $|X| > 1$, the MVS may be stored in the database system catalog either as multidimensional histograms, index statistics, or some other form of column-group statistics or statistics on intermediate tables. In practice, s_X is not known for all possible predicate combinations due to the exponential number of combinations of columns that can be used to define MVS. Suppose that s_X is known for every X in some collection[2] $T \subset 2^N$. Then the selectivity estimation problem is to compute s_X for $X \in 2^N \setminus T$.

It is intuitively clear that the query optimizer should avoid any extraneous assumptions about the unknown selectivities while simultaneously exploiting all existing knowledge in order to avoid unjustified bias towards any particular solution. In the Appendix, we survey the method that DB2 uses to compute missing selectivities and illustrate why this approach cannot use all existing knowledge without producing an inconsistent model. In the following section, we present the ME principle, which formalizes the notion of avoiding bias.

2.2 The Maximum-Entropy Principle

The maximum-entropy principle [GS85] models all that is known and assumes nothing about the unknown. It is a method for analyzing the available information in order to determine a unique epistemic probability distribution. Information theory [Sha48] defines for a probability distribution $\boldsymbol{q} = (q_1, q_2, \ldots)$ a measure of uncertainty called entropy: $H(q) = -\sum_i q_i \log q_i$. The ME principle prescribes selection of the unique probability distribution that maximizes the entropy function $H(\boldsymbol{q})$ and is consistent with respect to the known information.

[2] Note that the empty set $\emptyset$ is part of T, as $s_\emptyset = 1$ when applying no predicates.

Entropy maximization without any additional information uses the single constraint that the sum of all probabilities is equal to one. The ME probability distribution then is the uniform distribution. When constraints only involve marginals of a multivariate distribution, the ME solution coincides with the independence assumption.

Query optimizers that do not use MVS actually estimate their selectivities for conjunctive queries according to the ME principle: they assume uniformity when no information about column distributions is available, and they assume independence because they do not know about any correlations. By integrating the more general concept of maximum entropy into the optimizer's selectivity model, we thereby generalize the concepts of uniformity and independence. This enables the optimizer to take advantage of all available information in a consistent way, avoiding inappropriate bias towards any given set of selectivity estimates.

3. Consistent Selectivity Estimation Using Maximum-Entropy

The ME principle applied to selectivity estimation means that, given several selectivities of simple predicates and conjuncts, we choose the most uniform/independent selectivity model consistent with all of this knowledge.

3.1 The Constrained Optimization Problem

For each predicate p, write $p^1 = p$ and $p^0 = \neg p$. An *atom* is a term in disjunctive normal form (DNF) over the space of n predicates, i.e., a term of the form $\wedge_{i\in N} p_i^{b_i}$ for $b_i \in \{0,1\}$. We use $b = (b_1,\ldots,b_n) \in \{0,1\}^n$ to denote the atom $\wedge_{i\in N} p_i^{b_i}$. As a further abbreviation, we sometimes omit the parentheses and commas when denoting a specific atom. For example, using $P = \{p_1, p_2, p_3\}$ with $|P| = 3$, we denote by *100* the vector (1,0,0) and thus the atom $p_1 \wedge \neg p_2 \wedge \neg p_3$.

For each predicate p_X, $X \in 2^N$, denote by C(X) the set of components of X, i.e, the set of all atoms contributing to p_X. Formally,

$C(X) = \{b \in \{0,1\}^n \mid \forall\, i \in X : b_i = 1\}$ and

$C(\emptyset) = \{0,1\}^n$

For the predicates p_1 and $p_{1,2}$ we obtain:

$C(\{1\}) = \{100, 110, 101, 111\}$

$C(\{1,2\}) = \{110, 111\}$.

Additionally, for every $T \subseteq 2^N$, we denote by $P(b,T)$ the set of all known $X \in T \subseteq 2^N$ such that p_X has b as an atom in its DNF representation, i.e., $P(b,T) = \{X \in T \mid \forall\, i \in X : b_i = 1\} \cup \{\emptyset\}$. For the atom *011* and $T = 2^{\{1,2,3\}}$ we obtain the set $P(b,T) = \{\{2\}, \{3\}, \{2,3\}, \emptyset\}$.

Let x_b denote the selectivity of an atom, with $x_b \geq 0$. Given s_X for $X \in T$, we compute s_X for $X \notin T$ according to the ME principle. To this end, we must solve the following *constrained optimization problem*:

$$\text{minimize} \sum_{b\in\{0,1\}^n} x_b \log x_b \qquad (1)$$

given the |T| constraints

$$\sum_{b\in C(X)} x_b = s_X \quad \text{for all } X \in T \qquad (2)$$

The constraints are the known selectivities. The solution is a probability distribution with the maximum value of uncertainty (entropy), subject to the constraints. One of the included constraints is $s_\emptyset = \sum_{b\in\{0,1\}^n} x_b = 1$, which asserts that the combined selectivity of all atoms is 1. We can solve the above problem analytically only in simple cases with a small number of unknowns. In general, a numerical method is required.

3.2 Example

Figure 2 shows the probability space created for the predicate space created by $N = \{1,2,3\}$ and the knowledge set $T = \{\{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \emptyset\}$ with the selectivities $s_1 = 0.1$, $s_2 = 0.2$, $s_3 = 0.25$, $s_{12} = 0.05$, $s_{13} = 0.03$, and $s_\emptyset = 1$.

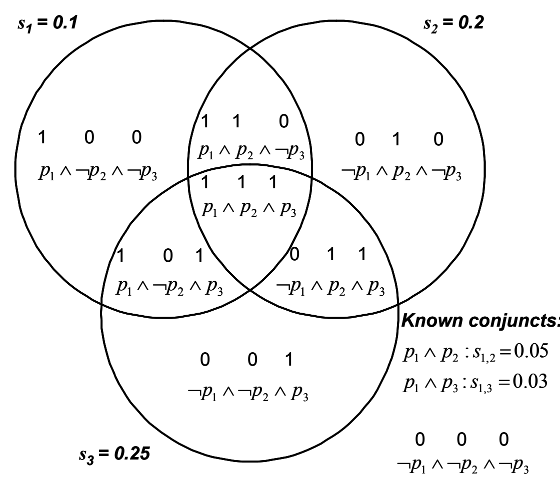

Figure 2: Probability space for |*T*|= 6 and N = {1,2,3}

This example results in the following six constraints:

s_1	$= x_{100} + x_{110} + x_{101} + x_{111}$	$= 0.1$	(I)
s_2	$= x_{010} + x_{011} + x_{110} + x_{111}$	$= 0.2$	(II)
s_3	$= x_{001} + x_{011} + x_{101} + x_{111}$	$= 0.25$	(III)
$s_{1,2}$	$= x_{110} + x_{111}$	$= 0.05$	(IV)
$s_{1,3}$	$= x_{101} + x_{111}$	$= 0.03$	(V)
$s_\emptyset$	$= \sum_{b\in\{0,1\}^3} x_b$	$= 1$	(VI)

The task of selectivity estimation is to now compute a solution for all atoms x_b, $b \in \{0,1\}^3$ that maximizes the entropy function $-\sum_{b\in\{0,1\}^3} x_b \log x_b$ and satisfies the above six constraints. All s_i, $i \in 2^{\{1,2,3\}}$, can then be computed from the x_b using the formula (2).

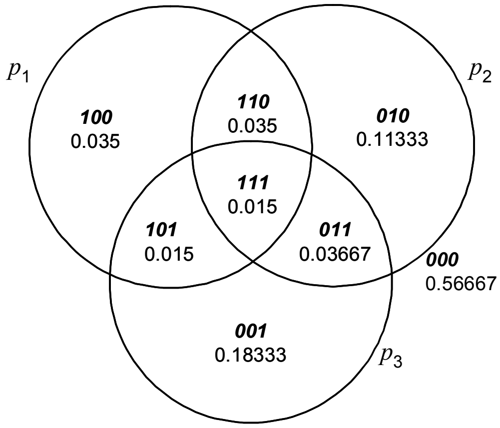

Figure 3: Maximum-Entropy Solution

Figure 3 gives the results obtained when solving this constrained optimization problem. For instance, in this ME solution, we obtain the selectivity estimate $s_{1,2,3} = x_{111} = 0.015$ and $s_{2,3} = x_{111} + x_{011} = 0.05167$.

In the next section, we describe an algorithm to compute this result efficiently for an arbitrary number n of simple predicates P and an arbitrary set T of conjunctions over simple predicates.

3.3 The Iterative Scaling Algorithm

To solve the constrained optimization problem in its general form, we first use the method of Lagrange multipliers to obtain a system of optimality equations. Since the entropy function is concave, a solution of this system yields the unique solution of the optimization problem [DR72]. We associate a Lagrange multiplier λ_X with each known s_X. This includes $\lambda_\varnothing$, a multiplier associated with the constraint $s_\varnothing = 1$. The Lagrangian for $\boldsymbol{x} = (x_b : b \in \{0,1\}^n)$ and $\boldsymbol{\lambda} = (\lambda_X : X \in T)$ is given by

$$L(\boldsymbol{x},\boldsymbol{\lambda}) = \sum_{b\in\{0,1\}^n} x_b \log x_b - \sum_{X\in T} \lambda_X \left(\sum_{b\in C(X)} x_b - s_X \right)$$

Taking derivatives with respect to the x_b and setting them equal to zero yields the optimality equations:

$$\log x_b + 1 = \sum_{X\in P(b,T)} \lambda_X . \qquad (3)$$

By making the substitution $z_X = e^{\lambda_X}$, we obtain an equivalent exponential form:

$$x_b = \frac{1}{e} \prod_{X\in P(b,T)} z_X . \qquad (4)$$

Using equation (4) to eliminate x_b in equation (2), we find that

$$\sum_{b\in C(Y)} \prod_{X\in P(b,T)} z_X = s_Y * e . \qquad (5)$$

It is usually not possible to analytically compute the Lagrange multipliers $Z = (z_1,\ldots, z_{|T|})$ satisfying equation (5). We use a variant of the iterative scaling algorithm [DR72] to efficiently obtain an approximate ME solution. The algorithm's main objective is to produce a new set Z of multipliers by iteratively refining the old Z^0. Each iteration consists of $|T|$ steps, during each of which the algorithm selects one $Y \in T$ and changes only the multiplier z_Y associated with the constraint induced by s_Y, while keeping the other multipliers constant. Therefore, temporarily the current equation (5) is satisfied for Z.

From equation (5) we can factor out the common Lagrange multiplier z_Y. Because this multiplier occurs in all summands on the left side, we can solve for z_Y to obtain the iterative scaling equation:

$$z_Y = \frac{s_Y * e}{\sum_{b\in C(Y)} \prod_{X\in P(b,T)\setminus\{Y\}} z_X} \qquad (6)$$

During each iteration step, the algorithm visits all members of Z, updating the value according to (6) for each z_Y, including $Y = \varnothing$. The sequence of iterated solutions converges to some limit Z [DR72]. The algorithm terminates when Δz, the change of all multipliers, becomes negligible between successive iterations. We then use Z to compute the values for the unknown x_b using equation (4), which, in turn, determines s_X for $X \notin T$. Figure 4 below shows the overall scaling algorithm.

```
Input:
        partial knowledge of a probability distribution,
        with known sY for every Y ∈ T ⊂ 2^N (∅ ∈ T)
Output:
        An approximate ME solution xb for all atoms b
        ∈ {0, 1}^n

FOR X ∈ T: zX := 1;  // ENDFOR X
ε := 10^-6;
Δz := ε; Δz^0 := 10 * ε;
WHILE abs(Δz^0 - Δz) > ε
  Δz^0 := Δz; Δz := 0;
  FOR Y ∈ T:
    sum := 0
    FOR b ∈ C(Y)
      product := 1
      FOR X ∈ P(b,T)\Y
        product *= zX; // ENDFOR X
      sum += product;  // ENDFOR b
    zY^0 := zY;
    zY := (sY * e) / sum;
  Δz += abs(zY^0 / zY); // ENDFOR Y
```

Figure 4: Iterative Scaling Algorithm

3.4 Integrating the Model into the DB2 Optimizer

The existing DB2 UDB optimizer precomputes selectivities from single-column statistics and MVS prior to plan enumeration, costing, and selection. When costing a plan that involves a conjunctive predicate p, the optimizer estimates p's selectivity by using a precomputed selectivity if available, or else combining several precomputed selectivities using the ad hoc method outlined in the Appendix. The optimizer takes precautions to avoid inconsistencies by keeping track of how a missing estimate for a predicate p was computed when it is first requested by a subplan during join enumeration. Subsequent requests for p's selectivity estimate will use this recorded information to ensure that the estimate is always derived in the same way. While this approach avoids inconsistencies, it does not utilize all available information as outlined previously and also causes an arbitrary bias towards particular plans, depending on how the selectivity was derived initially; this initial derivation depends in turn on the order in which selectivity estimates are requested during optimization.

The ME model can be seamlessly integrated into the DB2 UDB optimizer. For our prototype implementation, we extended the precomputation phase to not only compute selectivities based on statistical information, but also to precompute all missing selectivities using the iterative scaling algorithm. This precomputation eliminates the need to use the heuristics given in the Appendix for costing. It also eliminates the need to keep track of how selectivities were combined in order to avoid inconsistencies during estimation, because the ME estimates are inherently consistent with the statistics and generally applicable without causing any bias. The DB2 optimizer uses these precomputed selectivity estimates during dynamic programming to compute the cardinalities of each partial QEP it considers.

Our extensions to the cardinality model enable the optimizer to use all available statistics in a consistent way, for all plans in the plan space. This improved knowledge results in better query plans and improved query execution times, as shown experimentally in the next section. Our modifications also simplify the query optimizer's logic, as consistency checks and record-keeping are no longer necessary during cardinality estimation.

4. Experimental Evaluation

We applied the ME approach to estimate cardinalities for queries having conjunctive predicates on a single table. All available information about (joint) column frequencies was used and any unknown selectivities were estimated using ME.

To validate the quality of the ME estimates, we computed for each query the absolute error of the estimated cardinality relative to the true cardinality. We compared the - ME estimates to the state-of-the-art estimation method in DB2 UDB v8.2. Our test database is based on a 1GB real-world database and 200 queries from an actual Department of Motor Vehicles (DMV) workload. The workload involves the four tables OWNER, CAR, DEMOGRAPHICS, and ACCIDENTS. For all of our experiments (except the ones in Section 4.4), we focus on the portion of each query that references the CAR table. The schema of the CAR table is given in Figure 5.

id	Integer (Primary Key)
ownerid	Integer (Foreign Key)
year	Integer
make	Char (20)
model	Char (20)
color	Char (20)

Figure 5: Schema of the CAR table (DMV database)

The CAR table has a base cardinality of 143,309 rows. The table also contains strong correlations between MAKE, MODEL, COLOR, and YEAR; multivariate statistics are required on at least a subset of these columns in order to obtain reasonable cardinality estimates. We focused on the 3-way correlation between MAKE, MODEL, and COLOR, as this is the correlation most frequently encountered in actual queries.

4.1 Quality of the estimates

In order to assess the estimation quality of our ME approach, we ran experiments to estimate the selectivity of a query containing a conjunctive predicate formed by the three simple predicates:

p_1: CAR.MAKE = :literal1
p_2: CAR.MODEL = :literal2
p_3: CAR.COLOR = :literal3

*"Select * from car c where p1 AND p2 AND p3"*

We used a workload of 200 queries having different values for each of the three literals in each query instance. From the base statistics, we always knew the selectivities s_1, s_2, and s_3, of the simple predicates, i.e., $\{\{1\}, \{2\}, \{3\}\} \subseteq T$. Additionally, we created MVS to make the optimizer aware of several of the correlations between the three columns referenced by the predicates. We examined five major cases, each representing different available knowledge about joint column distributions. We label each case as "*f.c*," where *f* is the number of BFs in each known conjunct, and *c* is the number of known conjuncts:

Case 1.3: Only the selectivities of the single predicates (marginals) s_1, s_2, s_3 are known. In this special case, the ME solution can be computed analytically and is identical to Case I in the Appendix.

Case 2.1: Besides the marginals, the selectivity of one conjunctive predicate consisting of two BFs is known. Since our query comprises three BFs, we distinguish three possible subcases. In Case 2.1a, the selectivity $s_{1,2}$ for the conjunct on MAKE and MODEL is known. In Case 2.1b, the selectivity $s_{1,3}$ for the conjunct on MAKE and COLOR is known. In Case 2.1c, the selectivity $s_{2,3}$ for the conjunct on MODEL and COLOR is known. The ME estimate for $s_{1,2,3}$ is just the product $s_{1,2} * s_3$ for Case 2.1a, and analogously for the other cases. The structure of Case 2.1 coincides with that of Case III in the Appendix; the resulting ME solution can be computed analytically for this special case and is identical to the solution presented in the Appendix.

Case 2.2: Besides the marginals, the selectivity of two conjunctive predicates consisting of two BFs is known. In this case, we again distinguish three subcases, depending on which conjunctive predicates are known:
2.2a: $s_{1,2}$, $s_{1,3}$ for (MAKE, MODEL), (MAKE, COLOR)
2.2b: $s_{1,2}$, $s_{2,3}$ for (MAKE, MODEL), (MODEL, COLOR)
2.2c: $s_{1,3}$, $s_{2,3}$ for (MAKE, COLOR), (MODEL, COLOR)

In this special case, the ME solution can again be computed analytically. For Case 2.2a, $s_{1,2,3} = s_{1,2} * s_{1,3} / s_1$, and analogously for Case 2.2b and Case 2.2c. Note that this solution does *not* correspond to the computation described in Case IV in the Appendix.

Case 2.3: Statistics on all three conjunctive predicates consisting of two BFs are known. In this case, the ME problem can no longer be solved analytically, and the iterative scaling algorithm must be applied. Note that the ME result in this case again does not correspond to the computation described in Case IV in the Appendix.

Case 3.1: Given perfect knowledge of the conjunctive predicate we seek to estimate, we know the selectivity $s_{1,2,3}$ of the overall conjunctive predicate consisting of three BFs. The ME solution in this case returns $s_{1,2,3}$ as the answer, which corresponds to Case II in the Appendix.

Using All Information

For a workload of two-hundred queries, the box plot in Figure 6 shows the absolute estimation error for all of the cases discussed above when estimating the selectivity of a conjunctive predicate on MAKE, MODEL, and COLOR. The bottom of the box shows the first quartile of the absolute estimation error experienced for the workload. The line inside the box and the number to the right of the line show the value of the median. The top of the box gives the third quartile, and the line above the box gives the maximum value. Because there are strong correlations, Case 1.3, which uses the independence assumption, exhibits considerable estimation errors, with a median of 788 and a maximum error of almost 10,000. Using knowledge about correlation between one pair of columns reduces both the maximum and median error by one order of magnitude. Observe that the correlation between MAKE and MODEL is apparently stronger than the correlation between the other pairs, since knowing about this correlation results in a smaller error (median of 42 and maximum of 692) relative to knowing any other single pair.

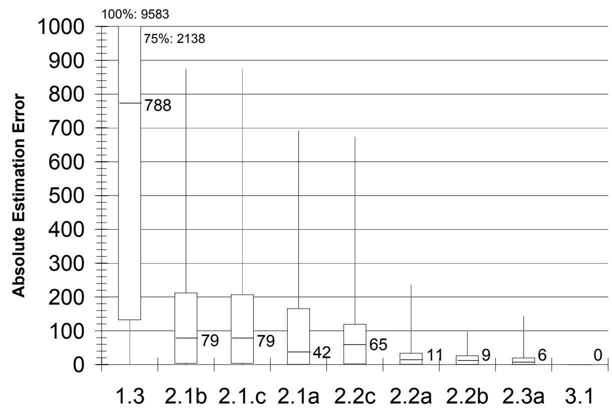

Figure 6: Box Plot comparing the Absolute Estimation Error for various |*T* | (200 queries)

If two pairs of correlations are known, then the error is reduced even further: if one of the pairs is the strong correlation between MAKE and MODEL, the median error is reduced by two orders of magnitude over Case 1.3 (pure independence). The maximum error in this case also gets reduced by almost two orders of magnitude. Note that when we only know the not-so-strong (MAKE, COLOR) and (MODEL, COLOR) correlations in Case 2.2c, the overall error is still reduced over only knowing one of the pairs, but the reduction (for instance, from 79 to 65 for the median) is not as pronounced as when the correlation on (MAKE, MODEL) was known, in which case the median is reduced from 79 down to 11 and 9.

Both the median and 75th percentile of the absolute error are reduced even further in Case 2.3, where three conjuncts are known (e.g., the median being reduced down to 6 in this case). In Case 3.1, perfect knowledge results in zero error.

Overall, the experiment shows that our ME estimation method has the desired property of reducing the absolute estimation error as more knowledge becomes available. We conducted further experiments for queries involving more than three predicates, with similar results; we omit these experiments due to space limitations.

Improvement over the State of the Art

Figure 7 contrasts the absolute estimation error of our ME approach with the state-of-the-art (SOTA) method in cardinality estimation, as outlined in the Appendix. Note that the estimation errors of SOTA and ME are identical for the Cases 1.3, 2.1, and 3.1, because the ME principle reduces to the assumptions of uniformity and independence that SOTA uses in these cases. However, as soon as several pieces of statistical information about one column are available, SOTA and ME return different results. In these cases, SOTA cannot use the additional information to improve estimates, and thus is forced to use only information about one conjunct.

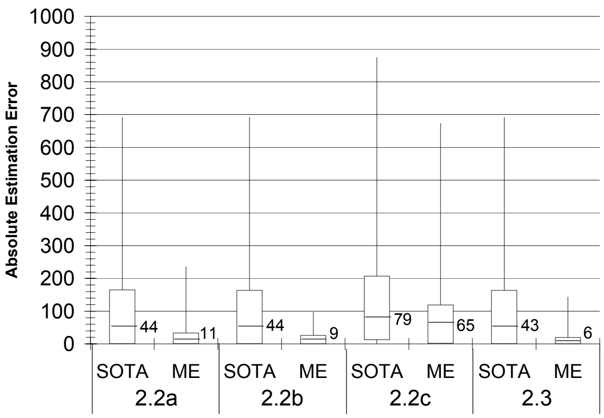

Figure 7: Comparison of ME to SOTA (200 queries)

In contrast, ME does not apply erroneous independence assumptions, but uses all available information to reduce the estimation error substantially. If one of the two known pieces of information is the strong correlation between MAKE and MODEL (Cases 2.2a and 2.2b), then ME reduces the median error by a factor of 4 and the worst-case error by more than one order of magnitude. Comparing Cases 2.1b and 2.1c in Figure 6 with Case 2.2c in Figure 7 shows that combining the two weaker correlations (MAKE, COLOR) and (MODEL, COLOR), is a better strategy than picking any single piece of correlation information: the median error is reduced from 79 to 65. If all three correlations are known (Case 2.3), then ME improves over SOTA by almost one order of magnitude in median, and by more than two orders of magnitude in the worst case.

Figures 6 and 7 together show that ME generalizes the optimizer's principles of uniformity and independence: ME yields the same estimates as SOTA if information is available only about single columns or only about a single conjunct. As soon as statistical information about more than one conjunct is available, ME exhibits superior performance by combining all of the information. The improvement in estimation quality over SOTA increases as the number $|T|$ of pieces of available information increases.

We have conducted further experiments for more than three BFs. These experiments confirm what we illustrate above for the case $|P| = 3$: the more the available information, the greater the improvement of the ME estimates over the SOTA estimates. For larger $|P|$ and larger $|T|$, the improvement of ME over SOTA quickly becomes several orders of magnitude, in both median and worst-case absolute error.

4.2 Computational Cost

Iterative scaling, as described in Section 3.3, is exponential in the number of predicates and linear in the number of constraints. Therefore, it is important to analyze the maximum number of predicates for which this algorithm is feasible in practice.

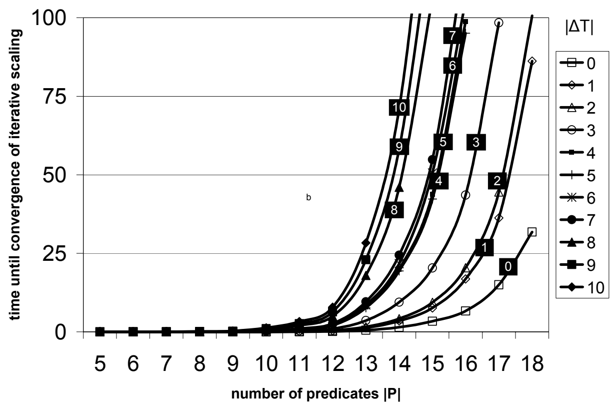

Figure 8: Time needed to solve the constrained optimization problem for various $|P|$ and $|\Delta T|$

Figure 8 shows the elapsed time in seconds until iterative scaling converged, depending on the number of simple predicates $|P|$ and the amount $|\Delta T|$ of available knowledge about conjuncts over P, i.e., $\Delta T = T \setminus N$. The time for $|P| < 5$ was less than one second for all configurations of $|T|$, and thus is omitted from the figure. We observe that iterative scaling exhibits sub-second convergence time on a laptop using an Intel Pentium(R) III Mobile CPU 1133MHz with 512MB RAM, as long as $|P|$ is below 8. The performance impact of $|T|$ is negligible.

For larger numbers of predicates, the exponential nature of the iterative scaling algorithm drastically increases the response time, making this algorithm impractical for more than ten predicates, as the response time then exceeds one second, thus having a clearly noticeable impact on the overall query optimization time. Moreover, when $|P|$ is large the amount of available knowledge $|T|$ has a noticeable impact on the overall performance. Fortunately, the algorithm's performance was generally acceptable when iterative scaling was performed over the local predicates on each single table separately (i.e., P being the set of local predicates on a table). We rarely encountered more than ten local predicates on a single table, even for very complex real-world queries: All but five queries in the customer workloads available to the authors had less than 6 local predicates on a single table, while the total number of predicates in many queries exceeded 50. Only three queries had more than eight predicates on a single table, 11 being the maximum. As discussed in Section 5, when the number of predicates exceeds 8 we can often use preprocessing steps to bring down the number of predicates used for scaling to a reasonable number.

4.3 Backward Computation

We can also use the iterative scaling algorithm for backward computation, i.e., using knowledge of s_X to compute s_Y, where $Y \subset X$. This means that if we have the three-column correlation between MAKE, MODEL, and COLOR, the ME approach allows us to exploit that information correctly rather than assuming independence for any of the two-column correlations between MAKE and MODEL, MAKE and COLOR, or MODEL and COLOR. Figure 9 displays box plots of the absolute estimation error for the SOTA method versus the ME approach when computing the cardinalities for this example.

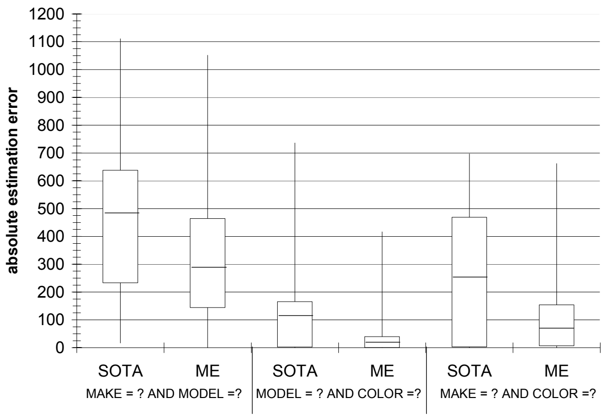

Figure 9: Comparison of ME and SOTA when estimating $s_{1,2}$, $s_{2,3}$, or $s_{1,3}$ from $s_{1,2,3}$ (200 queries)

As can be seen, use of the ME method can reduce the estimation error in some situations by an order of magnitude. Use of ME reduces the median estimation error by more than a factor of 2 when estimating MAKE and MODEL, by a factor of 6 when estimating MODEL and COLOR, and by factor of 4 when estimating MAKE and COLOR. Note that the reduction is smaller for MAKE and MODEL than for the other cases, as ME – when knowing nothing about the two-way correlations – distributes the correlation "uniformly" among all three two-way correlations. Because the true correlation between MAKE and MODEL is the strongest of the various correlations, the correction achieved by ME in this case is smaller than in the other two cases.

4.4 Query Execution Time

Although improving the cardinality estimates yields a better model for the query optimizer, the bottom line of optimization is the improvement in query execution time. We measured the impact that the ME method has on the query execution time for our example workload of 200 queries against the DMV database. Recall that, in the previous experiments, we used only the part of each query that selects a particular MAKE, MODEL, and COLOR from the CAR table, ignoring all other tables referenced by the query. In this experiment we used each query in its entirety. Besides applying selection predicates to the CAR table, each full query also joins the CAR table with up to three additional tables and applies additional local predicates on the other tables. For all queries, the execution time of iterative scaling was less than one second and also below 1% of the total query execution time.

Figure 10 shows the performance benefit observed when running the queries with MVS created for all three two-way correlations on CAR, improving the estimates for the CAR table as in Case 2.3 in Section 4.1. Of the 200 queries, 92 showed only marginal performance gains or no gains at all. On the other hand, many queries executed between two and five times faster when using the ME estimates as opposed to the SOTA estimates. Nine queries had a performance gain of more than one order of magnitude, due to improvements in join order and/or join methods; these improvements resulted from the better cardinality estimates on just the CAR table alone.

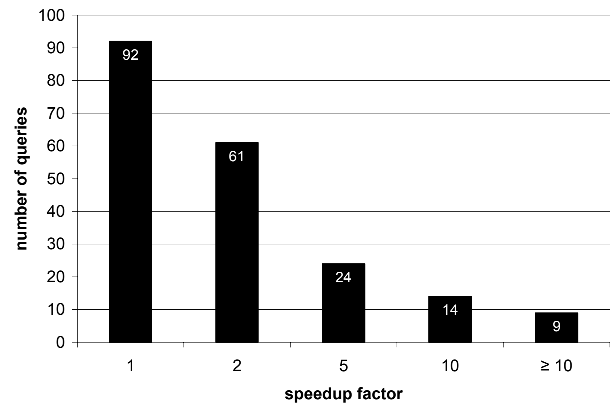

Figure 10: Performance benefit using two-way MVS

5. Some practical considerations

Even though the iterative scaling algorithm has a complexity of $O(|T| \cdot 2^{|P|})$, we often do not have to execute scaling for the entire predicate set P. In many situations, we can compute the maximum entropy solution by partitioning P into several independent subsets and by executing the scaling algorithm on each subset. This approach reduces the complexity substantially. In the following, we discuss several partitioning strategies that make the iterative scaling algorithm feasible for even extremely complex queries.

Complexity Reduction by Creating Relevant Partitions

P can be partitioned into non-overlapping subsets and simplified, depending on the available knowledge T, reducing the dimensionality of the problem space and thus the complexity of the scaling algorithm. Suppose that (a) for $U \subseteq N$, s_U is known or can be computed (possibly with closed formulas) in a way consistent with the ME principle using only a subset of the information in T, and (b) the selectivities s_X with $X \subseteq U$ are not important[3]. Then it is possible to replace the predicates denoted by U in P with a single predicate, reducing $|P|$, $|T|$, and the overall complexity of the problem. Below we provide several examples of when this is possible.

If T does not contain any information about a subset U of N, except for the single selectivities (i.e., $|X| = 1$ for all $X \in U$), then the selectivity s_U can be computed as $s_U = \Pi_{i \in U} s_i$. All simple predicates p_i, $i \in U$, can then be removed from P, and all pieces of knowledge in T that intersect U can be removed, creating a new set of knowledge T^*. Adding a new single predicate U with known selectivity thus reduces the complexity of iterative scaling to $O(|T^*| \cdot 2^{|P|-|U|+1})$. Similarly, for $U \in T$ and $|U| > 1$, all simple predicates p_i, $i \in U$, can be removed from P and replaced by a new single predicate p_U with known selectivity, again reducing the complexity of iterative scaling to $O(|T^*| \cdot 2^{|P|-|U|+1})$. Also, for the special case in which S contains knowledge as described in Section 4.1, Case 2.2, the selectivity of U can be computed as described there, and the complexity can be reduced to $O(|T^*| \cdot 2^{|P|-2})$.

The general approach to reducing complexity uses a preprocessing step in which the predicate space P is partitioned into independent components of nonintersecting knowledge based on both T and additional information about the relevance of the selectivities within each component. The selectivity of each component is computed either by using closed formulas or running iterative scaling; the latter process is fast because each component has relatively low complexity.

Complexity Reduction by Discovery of Implications

It is sometimes possible to identify a priori certain atoms b for which the input knowledge T implies that the selectivity x_b is zero. Iterative scaling can then exclude these atoms from the iteration step (8) in the algorithm. There are two ways to deduce from T that $x_b = 0$ for some $b \in \{0,1\}^n$.

(a) If $s_x = 0$ for some $X \in T$, then $x_b = 0$ for all atoms $b \in C(X)$. None of these atoms need to be considered by the iterative scaling algorithm.

(b) If two sets of atoms A and B with $B \subset A$ have the same selectivity, i.e., $\Sigma_{b \in B} x_b = \Sigma_{b \in A} x_b$, , then $x_b = 0$ for $b \in A \backslash B$, and these atoms can be ignored as well.

We conducted several experiments after implementing our techniques for reducing the complexity of the scaling algorithm, the details of which we omit due to length restrictions. In summary, the experiments showed that these simple pre-processing steps significantly reduce the number of iterations needed for convergence. For several real-world queries and real-world data, the running time was improved by orders of magnitude, producing ME distributions for more than twenty predicates on a single table with sub-second execution times.

[3] s_X with $X \subseteq U$ usually is only important when a query plan exists that needs s_X. For an optimizer considering plans with multiple predicates on a single table, this typically is only the case if an index on the columns in X exists.

6. Related Work

Cost-based query optimization was introduced in [SAC+79]. Many researchers have investigated the use of statistics for query optimization, especially for estimating the selectivity of single-column predicates using histograms [PC84, PIH+96, HS95] and for estimating join sizes [Gel93, IC91, SS94] using parametric methods [Chr83, Lyn88]. Some MVS proposed for cardinality estimation include multidimensional histograms [PI97], statistics on views [GJW+03, BC03], and Bayesian [GTK01] or other forms [DGR01] of probabilistic models.

Recent research in selectivity estimation has mostly focused on determining which statistics to collect and how to collect them efficiently. The methods in [BC02, CN00] analyze the query workload to select a set of statistics to maintain, such as MVS on base data or query expressions. The paper [SLM+01] employs a feedback loop in which query executions are monitored, the monitored information is analyzed to determine estimation errors, and the feedback information is used to adjust various stored statistics. The approach of [AC99, BC01] maintains MVS based on query feedback by incrementally building a multidimensional histogram that can be used to estimate the selectivity of conjunctive predicates. The paper [AHL+04] uses runtime feedback and other techniques to determine when and how to collect statistics. [IMH+04] use a sampling based chi-squared test to determine the most relevant MVS to collect for query optimization.

Although a large body of work, as surveyed above, focuses on various aspects of recommending, storing, and maintaining statistics for query optimization, no previous research has addressed the problem of combining selectivities derived from multiple available MVS to improve selectivity estimation of conjunctive predicates. Past work either has derived selectivities for each conjunct from single-column statistics and combined them using the assumption of statistical independence, or has used ad hoc techniques to determine the "best" of many competing MVS for estimating the selectivity of conjunctive predicates [BC02, GJW+03, BC04].

Information theory and the broad mathematical principle of maximum entropy have been applied to other domains such as machine translation [GON+01] and information retrieval [CWC91].

7. Conclusions

We have presented a novel ME method for estimating the selectivity of conjunctive predicates, The method is based on an information-theoretically sound approach that takes into account available statistics on both single columns and groups of columns. The model avoids arbitrary biases, inconsistencies, and the flight from knowledge to ignorance by deriving missing knowledge using the ME principle. This principle consistently extends the principles of uniformity and independence used in state-of-the-art selectivity models to exploit any available multi-attribute information. We have described, and analyzed a specific embodiment of our approach that solves the constrained ME optimization using Lagrange multipliers and an iterative scaling algorithm. We have implemented this method in a prototype version of DB2 UDB, improving the quality of the query optimizer while also simplifying its logic.

The ME approach improved selectivity estimates significantly over the state-of-the-art model used in the commercial DBMS, often by orders of magnitude. This in turn resulted in a considerable improvement of query execution times for an example workload on a DMV dataset derived from a real-world database and workload. Several queries ran orders of magnitude faster. Iterative scaling produced selectivity estimates for all practical queries with sub-second response times, adding less than a second to optimization time in all but the most complex queries.

Future work includes the investigation of efficient alternatives to the iterative scaling algorithm to further reduce the time required to compute the ME solution; we are currently investigating an algorithm based on Newton's method. Furthermore, we are working on extending the scope of our ME method to permit cardinality estimation for join predicates and distinct projections; the latter functionality is needed for optimizing queries with DISTINCT or GROUP BY clauses.

References

AC99 A. Aboulnaga, S. Chaudhuri: Self-tuning Histograms: Building Histograms Without Looking at Data. SIGMOD 1999: 181-192

AHL+04 A. Aboulnaga, P. Haas, S. Lightstone, et al.: Automated Statistics Collection in DB2 UDB. VLDB 2004: 1146-1157

ATL+03 M. Ault, M. Tumma, D. Liu, et al.. Oracle Database 10g New Features: Oracle10g Reference for Advanced Tuning and Administration. Rampant TechPress, 2003.

BCG01 N. Bruno, S. Chaudhuri, L. Gravano: STHoles: A Multidimensional Workload-Aware Histogram. SIGMOD 2001

BC02 N. Bruno and S. Chaudhuri: Exploiting Statistics on Query Expressions for Optimization. SIGMOD 2002

BC03 N. Bruno, S. Chaudhuri: Efficient Creation of Statistics over Query Expressions. ICDE 2003:

BC04 N. Bruno, S. Chaudhuri: Conditional Selectivity for Statistics on Query Expressions. SIGMOD 2004: 311-322

Chr83 S. Christodoulakis: Estimating record selectivities. Inf. Syst. 8(2): 105-115 (1983)

CN00 S. Chaudhuri, V. Narasayya: Automating Statistics Management for Query Optimizers. ICDE 2000: 339-348

CWC91 D. Chiu, A. Wong, B. Cheung: Information Discovery through Hierarchical Maximum-entropy Discretization and Synthesis. Knowledge Discovery in Databases 1991: 125-140

DGR01 A. Deshpande, M. Garofalakis, R. Rastogi: Independence is Good: Dependency-Based Histogram Synopses for High-Dimensional Data. SIGMOD 2001

DR72 J.N. Darroch and D. Ratcliff: Generalized iterative scaling for log-linear models. The Annals of Mathematical Statistics (43), 1972:1470–1480.

DR99 D. Donjerkovic and R. Ramakrishnan: Probabilistic Optimization of Top N Queries. VLDB 1999: 411-422

Gel93 A. Van Gelder: Multiple Join Size Estimation by Virtual Domains, PODS 1993.

GJW+03 C. Galindo-Legaria, M. Joshi, F. Waas, et al: Statistics on Views. VLDB 2003: 952-962

GON+01 I. García-Varea, F. Och, H. Ney, et al: Refined Lexikon Models for Statistical Machine Translation Using a Maximum-entropy Approach. ACL 2001: 204-211

GP00 W. Greiff, J. Ponte: The maximum-entropy approach and probabilistic IR models. ACM TIS. 18(3): 246-287, 2000

GS85 S. Guiasu and A. Shenitzer: The principle of maximum-entropy. The Mathematical Intelligencer, 7(1), 1985.

GTK01 L. Getoor, B. Taskar, D. Koller: Selectivity Estimation using Probabilistic Models. SIGMOD 2001

HS95 P. Haas and A. Swami: Sampling-Based Selectivity Estimation for Joins Using Augmented Frequent-Value Statistics. ICDE 1995.

IBM02 DB2 Universal Database for iSeries: Database Performance and Query Optimization. IBM Corp., 2002.

IBM04 DB2 v8.2 Performance Guide. IBM Corp., 2004.

IC91 Y.E.Ioannidis and S.Christodoulakis: Propagation of Errors in the Size of Join Results. SIGMOD 1991

IMH+04 I. F. Ilyas, V. Markl, P. J. Haas, P. G. Brown, A. Aboulnaga: CORDS: Automatic discovery of correlations and soft functional dependencies. Proc. 2004 ACM SIGMOD, June 2004.

Lyn88 C. A. Lynch: Selectivity Estimation and Query Optimization in Large Databases with Highly Skewed Distribution of Column Values. VLDB 1988: 240-251

Mic04 SQL Server 2000 Books Online v8.00.02. Microsoft Corp., 2004.

PC84 G. Piatetsky-Shapiro, C. Connell: Accurate Estimation of the Number of Tuples Satisfying a Condition. SIGMOD Conference 1984: 256-276

PIH+96 V. Poosala, et. al: Improved histograms for selectivity estimation of range predicates. SIGMOD 1996

PI97 V. Poosala and Y. Ioannidis: Selectivity Estimation without value independence. VLDB 1997

SAC+79 P.G. Selinger et al: Access Path Selection in a Relational DBMS. SIGMOD 1979

Sha48 C. E. Shannon: A mathematical theory of communication, Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656, July and October, 1948

SLM+01 M. Stillger, G. Lohman, V. Markl, and M. Kandil: LEO – DB2's Learning Optimizer. VLDB 2001

SS94 A. N. Swami and K. B. Schiefer,:On the Estimation of Join Result Sizes. EDBT 1994

Acknowledgements

We would like to thank Kevin Beyer as well as our 2002 and 2003 interns, Kwong Yung and Amol Deshpande, for thoughts and help on early investigations of how to apply information theory and the principle of maximum entropy to selectivity estimation.

Appendix: Cardinality Estimation for Conjunctive Predicates with MVS in DB2

We describe the state of the art in exploiting MVS during query optimization. To our knowledge, prior research has neither formalized nor described solutions to this problem that go beyond classical independence and using MVS when they exactly match the columns referenced in a conjunctive predicate. We therefore describe the approach currently taken by DB2 UDB. In the absence of a complete set of statistical information, and due to the complexity and the cost of collecting such information, the DB2 UDB optimizer tries to exploit as much statistical data as is available. We describe four different scenarios. In the following we assume that the selectivities of simple predicates are always known.

Case I: No statistics on conjunctive predicates are known.
For this trivial case, the selectivity of the conjunctive predicate is the product of the selectivities of the individual predicates.

Example: Estimate $s_{1,2,3}$ *given* s_1 , s_2 , *and* s_3.

$s_{1,2,3} = s_1 * s_2 * s_3$.

Case II: Statistics on the conjunctive predicates are known and there exists a conjunctive predicate whose BFs match the set of predicates whose combined selectivity is to be determined.
For this simple case, the selectivity of the full conjunctive predicate is the selectivity of the conjunctive predicate whose BFs match the set of predicates in question.

Example:
(a) *Estimate* $s_{1,2,3}$ *given* $s_{1,2,3}$.
$s_{1,2,3} = s_{1,2,3}$.
(b) *Estimate* $s_{1,2,3}$ *given* $s_{1,2}$, $s_{2,3}$, *and* $s_{1,2,3}$.
$s_{1,2,3} = s_{1,2,3}$.

Case III: Statistics on the conjunctive predicates are known and there exist some conjunctive predicates whose intersection is an empty set.
In this case, the selectivity of the conjunctive predicate is the product of the selectivities of all conjunctive predicates and the selectivities of the individual predicates not participating in any conjunctive predicate.

Example:
(a) *Estimate* $s_{1,2,3,4,5}$ *given* $s_{1,2}$ *and* $s_{3,4}$
$s_{1,2,3,4,5} = s_{1,2} * s_{3,4} * s_5$.
(b) *Estimate* $s_{1,2,3,4,5}$ *given* $s_{1,2,5}$ *and* $s_{3,4}$
$s_{1,2,3,4,5} = s_{1,2,5} * s_{3,4}$.

Case IV: Statistics on the conjunctive predicates are known and there exists some conjunctive predicates whose intersection is not an empty set.

In this case, the selectivity of the conjunctive predicate is the product of the selectivities of all conjunctive predicates with the highest degree of correlation and the selectivities of the individual predicates not participating in any conjunctive predicate.

A conjunctive predicate p_X has a higher degree of correlation than a conjunctive predicate p_Y if

a) the number of BFs in p_X is greater than in p_Y, or
b) the number of BFs in p_X is equal to that in p_Y and $s_X / \prod_{i \in X} s_i \geq s_Y / \prod_{i \in Y} s_i$, or
c) if still a tie, the optimizer will just arbitrarily pick one of the conjunctive predicates.

Example:
(a) *Estimate* $s_{1,2,3,4,5}$ *given* $s_{1,2}$ *and* $s_{2,3,4}$
$s_{1,2,3,4,5} = s_{2,3,4} * s_1 * s_5$

(b) *Estimate* $s_{1,2,3,4,5}$ *given* $s_{1,2}$ *and* $s_{2,3}$ *and* $s_{1,2}/(s_1 * s_2) \geq s_{2,3}/(s_2 * s_3)$

$s_{1,2,3,4,5} = s_{1,2} * s_3 * s_4 * s_5$
(c) *Estimate* $s_{1,2,3,4,5}$ *given* s_{12} , $s_{2,3}$, $s_{3,4}$ *and* $s_{4,5}$, *and*
$s_{1,2}/(s_1 * s_2) \geq s_{2,3}/(s_2 * s_3)$,
$s_{1,2}/(s_1 * s_2) \geq s_{3,4}/(s_3 * s_4)$ *and*
$s_{1,2}/(s_1 * s_2) \geq s_{4,5}/(s_4 * s_5)$

$s_{1,2,3,4,5} = s_{1,2} * s_3 * s_4 * s_5$

Scenarios as in Case IV are especially likely to produce large errors in practice. Indeed, as we can see in Examples IV.a, IV.b, and IV.c, the ad hoc method has some serious deficiencies:

(i) In Case IV(a), the final estimate ignores the correlation between s_1 and s_2. Likewise, in Case IV.c, the estimation method erroneously assumes that s_3, s_4, and s_5 are completely independent of each other.

(ii) Although the optimizer is provided with more information in Case IV(c) than in Case IV(b), the estimate in Case IV(c) does not improve over that in Case IV(b). In order to avoid bias when comparing multiple plans during dynamic programming, the optimizer is thus forced to ignore valuable additional information

Both of these problems motivate our new solution based on the ME model.