CASE-DB is a relational database management system that allows users to specify time constraints in queries. For an aggregate query \( \text{AGG}(E) \) where \( \text{AGG} \) is one of COUNT, SUM and AVERAGE, and \( E \) is a relational algebra expression, CASE-DB uses statistical estimators to approximate the query. This paper extends our earlier work on statistical estimators of CASE-DB with the following features: (a) new statistical estimators for COUNT queries with projection, (b) extending the methodology for SUM and AVERAGE aggregate queries, (c) new sampling plans based on systematic sampling and stratified sampling. We also present performance evaluation experiments of the estimators with the above extensions using correlated and uncorrelated database instances.

In real-time (or time-constrained) databases, queries have to be completed within a given time period. To give an example, a mutual fund manager may need a risk analysis report (consisting of figures derived from a database using statistical and probabilistic methods) in a very short time (e.g., 10 minutes) in order to make a buy/sell decision during a market opportunity. Or, in a factory environment, a computer operator may need within 10 seconds the names and average temperature values of factory furnaces with “dangerously high” temperature readings last night. The database system may judge that this query cannot possibly be evaluated in 10 seconds. When an aggregate query in such an environment cannot be evaluated within the given time period, one approach is to evaluate a statistical estimator, and produce a statistical estimate as an approximate answer to the query. Such an approach has been proposed for COUNT relational algebra queries (Lipton and Naughton, 1989; Lipton, Naughton and Schneider, 1990; Hou and Ozsoyoglu, 1991; Hou and Ozsoyoglu, 1993). We have implemented our approach in a disk-based prototype DBMS, called CASE-DB and its main memory version, called CASE-MDB.

Our earlier work on CASE-DB used only COUNT estimators, and simple random sampling in providing approximate answers to user queries. This paper extends and completes our work on statistical estimators for aggregate relational algebra queries in CASE-DB with new estimators for COUNT queries, extension of the methodology into SUM and AVERAGE queries, the use of stratified sampling and correlated data:

1. **New statistical estimators for COUNT(E) queries with projection.** In our earlier work, whenever the relational algebra (RA) expression \( E \) contained the projection operator, we used a revised version of the Goodman’s estimator (Goodman, 1949) which did not always perform well. In this paper, we introduce two new estimators, namely, the Jackknife estimator (Burnham and Overton, 1979), and the Chao’s estimator (Chao, 1984). Very recently, several additional estimators are proposed for \( \text{COUNT}(\pi (r)) \) queries (Haas, Naughton, Seshadri, and Stokes, 1995), and evaluated using highly skewed data.

2. **Extending the methodology for SUM and AVERAGE aggregate queries.** The previous works introduced estimators for COUNT queries. We now propose and evaluate estimators for SUM and AVERAGE queries. To obtain estimators for \( \text{SUM}(E) \) and \( \text{AVG}(E) \) queries, where \( E \) has a projection operator, we use the double sampling technique with the acceptance/rejection method.

3. **New sampling plans based on systematic sampling and stratified sampling.** To obtain samples for evaluating an estimator, earlier work used simple random sampling (Hou and Ozsoyoglu, 1991), simple random sampling with an adaptive stopping criteria (Lipton, Naughton and Schneider,
1990), and cluster sampling (Hou and Ozsoyoglu, 1991). We now add systematic sampling and stratified sampling into the list of possible sampling techniques for aggregate query evaluation.

4. Evaluating the estimators with correlated data. In our earlier work, we have evaluated our estimators with artificial relation instances where each column of a relation was independent (i.e., uncorrelated) from other columns. In this paper, we evaluate our estimators with relation instances whose columns pairwise have controlled correlation. We also give an algorithm to create such a relation instance. To the best of our knowledge, the issue of creating relation instances with controlled correlation among its columns has not been addressed in the literature, and, we think, is important in other experimental areas such as benchmarking or query processing in databases.

Section 2 introduces the new estimators for COUNT(π(r)) queries. Generalization into estimators for COUNT(E) is given later in section 6. Section 3 introduces the double sampling approach to estimate SUM(π(π(π(r)))) in section 4, we briefly summarize systematic sampling and stratified sampling. Section 5 discusses how to create relations whose columns have pairwise controlled correlations, and each column is normally distributed with a prespecified mean and variance. Finally, in section 6, we present the algorithms for evaluating estimators for AGG(E) where AGG {COUNT, SUM, AVE} and E is an arbitrary RA expression. Section 7 is the experimental results. Please note that full versions of the experimental results summarized in section 7 are published elsewhere (Liu, 1989; Tjahjana, 1990; Du, 1993).

New Estimators for COUNT

Consider the query COUNT(π(π(r))). Conceptually, the projection operation π(π(r)) eliminates from each tuple those attributes not in X (thus creating classes of identical tuples), reduces each class of identical tuples to a single tuple, and returns the resulting tuples. That is, the projection operation on r may produce (and later eliminate) duplicate tuples. In the case that the sizes of distinct classes produced by projection are different, the duplication makes the probabilities of inclusion in the sample unequal; hence, introducing bias to the estimate for COUNT(π(π(r))).

Goodman (1949) proposed a nonparametric estimation of the number of groups in a population with known size. In earlier work, Hou, Ozsoyoglu and Taneja, 1989), we used the Goodman’s estimator to estimate (π(π(r))). However, when the sampling fraction is low, Goodman’s estimator is unstable for a population with heavy duplicates.

Below we present two different nonparametric estimators to estimate COUNT(π(π(π(r)))), namely, the Jackknife estimator and the Chao’s estimator. Burnham and Overton (1979) use the Jackknife method to estimate the population size (e.g., in our case, π(π(r)) is the population, and COUNT(π(π(π(r)))) is the population size) when sample inclusion probabilities vary among population elements. Chao (1984) proposes another nonparametric method to estimate the number of classes in a population. In what follows, we briefly state the Goodman’s estimator for comparison purposes, and adapt the Jackknife and Chao’s estimators to estimate COUNT(π(π(π(r))))

Goodman’s Estimator

Goodman’s estimator, denoted by G* and based on simple random sampling is

$$G^* = \sum_{i=1}^{m} A_i x_i$$

(1)

where A_i is a function of N, m and i, x_i is the number of classes containing i elements (the number of resulting tuples in a class) in a sample of size m from relation r and N is the number of tuples in r.

Jackknife Estimators

The Jackknife estimator, denoted by J*, is developed by Burnham and Overton (1979) to be used for estimating the number of N distinct animals in live-trapping studies. The studies are done by trapping animals on c occasions. Capture frequencies, denoted by x_i for i=1,...,c, are then computed. Each x_i represents the number of distinct animals captured exactly i times, while x_0 is the number of distinct animals never trapped. Accordingly, d = Σ_i^n x_i is the number of individuals seen during the study. Thus, N = d + x_0 is equal to the size of the animal population (i.e., the number of distinct animals).

The animal population size estimation problem is mapped to our problem as follows: N is COUNT(π(π(π(E))); x_0 is the number of tuples which appear exactly i times in the sample; x_0 is the number of distinct tuples which are not seen in the sample. As an analogy for c in our problem, we could theoretically have taken c as the size of the original sample (yielding many zero values for the x_i’s). Equivalently, for convenience, we took c as the highest appearing frequency of sample elements. Therefore, in our problem: d = Σ_i^n x_i equals the number of distinct tuples seen in the sample, and N' = d + x_0 represents the number of distinct tuples in the population (i.e., COUNT(π(π(π(E))))).

The k^th order Jackknife estimator J_k* of J_k is defined as

$$J_k^* = d + \sum_{i=1}^{k} a_i x_i$$

(2)

where a_i = 0 for i>k.

From Burnham and Overton (1979), for any fixed value of c, the higher order Jackknife estimators (i.e., increasing k values) lead to greater bias reduction, but at the cost of increased sampling variance. Conversely, for any fixed value
Related Content

Normalizing Multimedia Databases
[www.igi-global.com/chapter/normalizing-multimedia-databases/11181?camid=4v1a](www.igi-global.com/chapter/normalizing-multimedia-databases/11181?camid=4v1a)

Managing Inconsistent Databases Using Active Integrity Constraints
[www.igi-global.com/chapter/managing-inconsistent-databases-using-active/11171?camid=4v1a](www.igi-global.com/chapter/managing-inconsistent-databases-using-active/11171?camid=4v1a)

Issues in Transaction-Time Temporal Object Database Systems
[www.igi-global.com/article/issues-transaction-time-temporal-object/3271?camid=4v1a](www.igi-global.com/article/issues-transaction-time-temporal-object/3271?camid=4v1a)

Transforming Activity-Centric Business Process Models into Information-Centric Models for SOA Solutions
[www.igi-global.com/chapter/transforming-activity-centric-business-process/63673?camid=4v1a](www.igi-global.com/chapter/transforming-activity-centric-business-process/63673?camid=4v1a)