Towards Estimating the Number of Distinct Value Combinations for a Set of Attributes

Xiaohui Yu*
Department of Computer Science
University of Toronto
Toronto, ON, M5S 3G4, Canada
xhyu@cs.toronto.edu

Calisto Zuzarte
IBM Toronto Lab
8200 Warden Avenue
Markham, ON, L6G 1C7, Canada
calisto@ca.ibm.com

Kenneth C. Sevcik
Department of Computer Science
University of Toronto
Toronto, ON, M5S 3G4, Canada
kcs@cs.toronto.edu

ABSTRACT

Accurately and efficiently estimating the number of distinct values for some attribute(s) or sets of attributes in a data set is of critical importance to many database operations, such as query optimization and approximation query answering. Previous work has focused on the estimation of the number of distinct values for a single attribute and most existing work adopts a data sampling approach. This paper addresses the equally important issue of estimating the number of distinct value combinations for multiple attributes which we call *COLSCARD* (for COLumn Set CARDinality). It also takes a different approach that uses existing statistical information (e.g., histograms) available on the individual attributes to assist estimation. We start with cases where exact frequency information on individual attributes is available, and present a pair of lower and upper bounds on COLSCARD that are consistent with the available information, as well as an estimator of COLSCARD based on probability. We then proceed to study the case where only partial information (in the form of histograms) is available on individual attributes, and show how the proposed estimator can be adapted to this case. We consider two types of widely used histograms and show how they can be constructed in order to obtain optimal approximation. An experimental evaluation of the proposed estimation method on synthetic as well as two real data sets is provided.

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Systems—*Query Processing, Relational Databases*

General Terms

Algorithms, Design, Experimentation

*Work done while visiting IBM Toronto Lab.

CIKM'05, October 31–November 5, 2005, Bremen, Germany.

Keywords

relational database, cardinality estimation

1. INTRODUCTION

Most query optimizers in relational database systems require cost estimation of various candidate execution plans for a given query in order to select a good one. Good plan costing can help avoid plans that are intolerably slow compared to better access plans. Cardinality estimation of the intermediate results is key to getting better plan cost estimates. The number of distinct values in an attribute or the number of value combinations in a set of attributes is critical for cardinality estimation of various relational operators. For example, the textbook formula (which is also widely used in commercial systems) for estimating the result size of $R_1 \bowtie_{R_1.a=R_2.a} R_2$, the join of two relations R_1 and R_2 on a certain attribute a, is $\frac{|R_1|*|R_2|}{max(V_1,V_2)}$, where $|R_1|$ and $|R_2|$ are the cardinalities of the two relations, and V_1 and V_2 are the numbers of distinct values in $R_1.a$ and $R_2.a$ respectively.

Besides its traditional role in query optimization, estimating the number of distinct values (or value combinations) is also of great value to fast approximate query answering [4], online aggregation [7], data exploration, among others. For instance, an online aggregation system might want to estimate the result size of a group-by query so as to show the users the percentage of groups that have been produced as the query is being processed.

Estimating the number of distinct values for a *single* attribute has been well-studied. It corresponds to the statistical problem of estimating the number of species or classes in a population [2] and a considerable number of estimators have been proposed. Extensive research has also been done in the database literature, either by adopting those estimators proposed in the statistics literature to the database setting (e.g., [8, 9]), or by proposing new improved estimators (e.g., [5, 3]). Almost all previous work has taken the sampling approach, in which the estimate is derived by taking a sample of the data. The fact that useful statistical information is commonly available on the data set (stored in the catalog in the forms of histograms, etc.) has been largely overlooked.

This paper is concerned with estimating the number of distinct value combinations for *a set of attributes* in a table (henceforth referred to as COLSCARD, which stands for COLumn Set CARDinality). Such estimation is required for many relational operations involving multiple attributes. For example, one might be interested in estimating the result cardinalities of the following queries because these queries may be part of more complex queries or be

needed for fast approximate answering purposes:

```
SELECT sales_date, sales_person,
       SUM(sales_quantity) AS units_sold
FROM sales
GROUP BY sales_date, sales_person
```

and

```
SELECT DISTINCT first_name, last_name
FROM authors
```

Current commercial databases may use attribute-group statistics or multiple-attribute index statistics for estimating distinct value combinations on multiple attributes. However, these statistics are not always available, and it is not easy to decide what combinations one should collect statistics on, because the number of possible combinations can be huge. When there is no statistical information on the groups of attributes that we are interested in, naive estimators currently used in commercial systems do not work well and often produce large errors.

Instead of extending the sampling approach from the single-attribute to the multi-attribute case, we attempt to improve the quality of estimation by making use of existing statistical information about individual attributes, which is usually maintained by the DBMS in the system catalog and readily available. Compared with the sampling approach, our approach does not require physical access to the data through disk I/O, and therefore tends to be much more efficient.

We start with the case where exact frequency information of all individual attributes involved is available. That is, we know exactly the number of times each value appears in each of the attributes. We first show that tighter upper/lower bounds can be achieved by exploiting the frequency information. We then proceed to propose an estimator based on probability theory. We also provide a sampling method to trade off the computation cost and accuracy of estimation.

For the more general case where the system can only afford to maintain partial frequency information (in the forms of histograms, etc.) because of resource constraints and other considerations, we show how approximate estimation can be made, and study when the smallest approximation error can be achieved by choosing the right set of frequencies to be stored in the histograms. We present optimality results for two widely used classes of histograms.

The rest of the paper is organized as follows. Section 2 discusses related work in greater detail. In Section 3, we formally define the problem and introduce the notations. In Section 4, we present results for the case where exact marginal frequencies on individual attributes are available. Section 5 deals with the more general case where only partial information is available, and presents optimality results for two classes of histograms. Experimental evaluation is done in Section 6, and we conclude the paper and discuss future work in Section 7.

2. RELATED WORK

Virtually all previous work on estimating the number of distinct values has been focused on ways to derive an estimate through sampling the data. There exists a sizable bibliography in the statistics literature on this problem, where statisticians developed various estimators for the purposes of estimating the number of species in an area or the number of classes in a population. Readers are referred to the survey by Bunge and Fitzpatrick [2] for further reference.

This problem has also been extensively studied in the database literature. Early work includes the results by Hou *et al.* [8, 9], Naughton and Seshadri [12], and Özsoyoglu [14], in which statistical estimation methods are developed with database application in mind; see Olken [13] for an excellent survey. Haas *et al.* [5] did an extensive study of existing estimators from the statistical literature and proposed several new sampling-based estimators, one of which is a hybrid estimator designed to take into account the degree of data skew, and is shown to outperform previous estimators over a wide range of attribute-value distributions. This result is further extended by Haas and Stokes [6], where the relationship between generalized jackknife estimators and other known estimators is revealed.

Estimating the number of distinct values through sampling alone is inherently difficult. Charikar [3] *et al.* establish a strong negative result showing that no estimator can guarantee small error across all possible attribute-value distributions with a reasonably small sample. They present an estimator called GEE (Guaranteed-Error Estimator), which is provably optimal in the sense that its worst-case error matches the lower bound of theoretical result within a small constant factor. Heuristic estimators are then presented to take advantage of the knowledge of the data skewness obtained through the sample.

Gibbons [4] considers a related problem, where *distinct samples* of the data are obtained and stored for fast yet approximate distinct values query answering. The main objective of this approach is to get a synopsis of the data specifically tailored for distinct values queries in order for such queries to be answered quickly (albeit approximately) with error guarantee without accessing the underlying data at query processing time. Unlike previous sampling-based methods where normally only a portion of the data is accessed, this approach requires a single scan of the data to build the synopsis.

All the aforementioned work does not consider the role of existing knowledge about the data, and simply relies on sampling (or data scan in the case of distinct samples) to obtain information from the data. Such knowledge can potentially be helpful for the estimation and is usually abundant in modern database systems taking the forms of histograms, indexes, partial data cubes, and so on. Aside from moving the focus from single attribute to a set of attributes, our work is complementary to the existing approaches in that we consider how to utilize the knowledge in a consistent way to obtain quality estimates.

3. PRELIMINARIES

Before proceeding to discuss specific algorithms, we first formally define the problem of estimating the number of distinct value combinations as follows.

Definition 1. number of distinct value combinations in a set of attributes Given a selected set of attributes $\mathcal{G} = (A_1, A_2, \ldots, A_m)$ in a relation R, the number of distinct value combinations for $\mathcal{G}$ is defined as the number of distinct tuples in $\pi_{\mathcal{G}}R$ (the projection of R on $\mathcal{G}$).

We use $d_i (1 \leq i \leq m)$ to denote the number of distinct values in attribute $A_i (1 \leq i \leq m)$ in $\mathcal{G}$, and use $D_{\mathcal{G}}$ to denote the number of distinct value combinations. Assuming an (arbitrary) total ordering on the values in each attribute, we denote by vector $\mathbf{v} = (v_1, v_2, \ldots, v_m)$ the value combination that takes on the v_1-th value in attribute A_1, the v_2-th value in attribute A_2, and so on. We denote by $\mathcal{V}$ the set of all value combinations present in $\pi_{\mathcal{G}}R$; hence $D_{\mathcal{G}} = |\mathcal{V}|$. Let $f_{\mathbf{v}}$ be the normalized frequency of the value combination $\mathbf{v}$ such that $\sum_{\mathbf{v}\in\mathcal{V}} f_{\mathbf{v}} = 1$.

4. ESTIMATING COLSCARD WITH KNOWN MARGINALS

The first case we consider is when the exact marginal frequencies are known, i.e., the frequency vector $\mathbf{f}_i$ for each attribute $A_i \in \mathcal{G}(i = 1, \ldots, m)$ is known. Here the frequency vector $\mathbf{f}_i = (f_{i1}, f_{i2}, \ldots, f_{id_i})(\sum_{j=1}^{d_i} f_{ij} = 1)$, with f_{ij} being the fraction of tuples having the j-th value of attribute A_i. As an example, if there are three distinct values in attribute A_i, and they appear in R 20, 40, and 140 times respectively, then $\mathbf{f}_i = (0.1, 0.2, 0.7)$.

When there exist functional dependencies among the attributes in $\mathcal{G}$, they can be effectively utilized to obtain accurate estimates. For example, if we would like to estimate the COLSCARD for the set of attributes (A_1, A_2), and there happens to exist a functional dependency $A_1 \rightarrow A_2$, then clearly the COLSCARD is equal to d_1, the number of distinct values in A_1. Since such estimation is very straightforward, we will not consider it further in this paper. Our particular focus will be on the general cases where no functional dependencies can be readily exploited.

4.1 Naive estimators

Assuming the data in different attributes are independently distributed, the total number of possible value combinations would be $\prod_{i=1}^{m} d_i$. One might be tempted to use it as an estimate of the COLSCARD. In fact, a naive estimator, which is currently used in the IBM® DB2® Universal Database™ product(DB2 UDB) and other commercial DBMS's, is as follows:

$$\hat{D_{\mathcal{G}}} = \min\left\{\prod_{i=1}^{m} d_i, N\right\} \quad (1)$$

where N is the size of the table, and serves as a sanity bound because $D_{\mathcal{G}}$ cannot be greater than the table size.

However, such an estimator is problematic in that it is just the *maximum* possible number of value combinations for $\mathcal{G}$; when $N = |R|$ is small, some less frequent value combinations may not appear in the relation at all.

EXAMPLE 1. *Suppose we need to estimate the COLSCARD for three attributes A_1, A_2, and A_3, and we know that $\mathbf{f}_1 = (0.1, 0.3, 0.6)$, $\mathbf{f}_2 = (0.01, 0.99)$, $\mathbf{f}_3 = (0.05, 0.95)$.*

Scenario 1: $N = 100$. One might want to estimate the number of possible value combinations to be $3 \times 2 \times 2 = 12$. However, the known $\mathbf{f}_i$'s imply that the probability of occurrence of the least possible value combination is $\mathbf{f}_{11} \times \mathbf{f}_{21} \times \mathbf{f}_{31} = 0.1 \times 0.01 \times 0.05 = 0.00005$, which indicates that such a value combination is very unlikely to appear in a relation of size $N = 100$. Therefore, the expected number of value combinations is expected to be less than 12.

Scenario 2: When N is less than $\prod_i d_i$, e.g., N = 11, we need to "truncate" the estimated number of possible combinations from 12 to 11, as it cannot be greater than N. The problem is that, with a relation of size 11, we do not expect some value combinations with small occurrence probabilities to appear (e.g., the above combination with probability 0.00005 and the one with probability $0.3 \times 0.01 \times 0.05 = 0.00015$). So again, the expected number of value combinations is expected to be less than the maximum possible number, 11.

As shown in the preceding example, using the naive estimator is often erroneous, and better estimates are desired. In the following subsection, we first show that tighter bounds are available on COLSCARD than the trivial bounds, and then discuss means by which a good estimate can be obtained.

4.2 Upper/Lower bounds on COLSCARD

Let us first consider the following question: what can be a lower and an upper bound on COLSCARD? Naturally, for the set of attributes $\mathcal{G} = (A_1, A_2, \ldots, A_m)$, COLSCARD is lower-bounded by $\max\{d_1, d_2, \ldots, d_m\}$, since it cannot be less than the number of distinct values in any of the constituent attributes. Moreover, as discussed earlier in this section, COLSCARD is upper-bounded by $\min\left\{\prod_{i=1}^{m} d_i, N\right\}$. In this paper, we call those two bounds the trivial bounds.

Are tighter bounds available? The answer is positive for the case of two attributes. Specifically, we have the following theorems

THEOREM 1. *For attribute set $\mathcal{G} = (A_1, A_2)$, we define $l_{ij} (i = 1, 2; j = 1, \ldots, d_i)$ as the minimum number of different values that have to be combined with f_{ij} given the marginals, i.e.,*

$$l_{ij} = \min\left\{|\mathcal{F}| : \mathcal{F} \subseteq \{1, \ldots, d_{i'}\}, S(\mathcal{F}) \leq f_{ij} < S(\mathcal{F}) + f_{i'q}, \forall q \notin \mathcal{F}\right\}$$

where $i' = 3 - i$, and $S(\mathcal{F}) = \sum_{p \in \mathcal{F}} f_{i'p}$. Then

$$D_{\mathcal{G}}^{\perp} = \max_{i=1,2}\left\{\sum_{j=1}^{d_i} l_{ij}\right\}$$

is a lower bound on $D_{\mathcal{G}}$, and $D_{\mathcal{G}}^{\perp} \geq \max\{d_1, d_2\}$.

THEOREM 2. *For attribute set $\mathcal{G} = (A_1, A_2)$, we define $u_{ij} (i = 1, 2; j = 1, \ldots, d_i)$ as the maximum number of different values that can be combined with f_{ij} given the marginals, i.e.,*

$$u_{ij} = \min\{d_{i'}, f_{ij} \cdot N\}$$

where $i' = 3 - i$. Then

$$D_{\mathcal{G}}^{\top} = \min_{i=1,2}\left\{\sum_{j=1}^{d_i} u_{ij}\right\}$$

is an upper bound on $D_{\mathcal{G}}$, and $D_{\mathcal{G}}^{\top} \leq \min\left\{\prod_{i=1}^{m} d_i, N\right\}$.

Theorems 1 and 2 provide tighter lower and upper bounds than the trivial ones, as illustrated in the following example.

EXAMPLE 2. *Consider the estimation of COLSCARD for $\mathcal{G} = (A_1, A_2)$, with N=10, $\mathbf{f}_1 = (0.4, 0.4, 0.2)$, and $\mathbf{f}_2 = (0.8, 0.1, 0.1)$. Clearly, the trivial lower and upper bounds are 3 and 9 respectively. The lower and upper bounds dictated by Theorems 1 and 2, on the other hand, are 4 and 5 respectively, which are tighter than the trivial bounds.*

Computing $D_{\mathcal{G}}^{\perp}$ can be done using the algorithm depicted in Algorithm 1, where for each f_{ij}, we calculate the corresponding l_{ij} by finding out at least how many frequencies from the other attribute have to be combined with it, and then add them up and get the final answer.

Obtaining the upper bound $D_{\mathcal{G}}^{\top}$ is much easier. For each f_{ij}, we compute u_{ij} by simply comparing $f_{ij} \cdot N$ and $d_{i'}$. Adding up the u_{ij}'s for each attribute and taking the lesser of the two sums give rise to the desired bound.

4.3 The expected number of value combinations

We now propose a method to estimate $D_{\mathcal{G}}$ based on probability theory. Let $\mathbf{f}_{\mathcal{G}} = \mathbf{f}_1 \otimes \mathbf{f}_2 \otimes \ldots \otimes \mathbf{f}_m$ be the Kronecker product (also known as direct product, tensor product, or outer product) of the frequency vectors of all attributes. Then $\mathbf{f}_{\mathcal{G}}$ is a vector of

Algorithm 1 Computing $D_{\mathcal{G}}^{\perp}$

```
sort f_i(i = 1, 2)in descending order;
for i = 1 to 2 do
  i' = 3 − i;
  for j = 1 to d_i do
    k = 1;
    while f_ij > Σ_{l=1}^{k} f_{i'l} do
      k = k + 1;
    end while
    l_ij = k;
  end for
  lb_i = Σ_{j=1}^{d_i} l_ij;
end for
D_G^⊥ = max_i{lb_i};
```

length $M = \prod_{i=1}^{m} d_i$. Note that each element of $\mathbf{f}_{\mathcal{G}}$ represents the frequency $f_{\mathbf{v}}$ of a specific value combination $\mathbf{v}$. Clearly, the individual components of the vector $\mathbf{f}_{\mathcal{G}}$ add up to 1, i.e., $\sum_{\mathbf{v}\in\mathcal{V}} f_{\mathbf{v}} = 1$. For instance, in the above example, $\mathbf{f}_{\mathcal{G}}$ = (0.00005, 0.00095, 0.00495, 0.09405, 0.00015, 0.00285, 0.01485, 0.28215, 0.0003, 0.0057, 0.0297, 0.5643), and it is easy to verify that they sum up to 1.

If we think of each element of $\mathbf{f}_{\mathcal{G}}$ as the probability of occurrence of the corresponding value combination, and each tuple in the relation R as independently drawn from M possible value combinations, then we have the following theorem for estimating COLSCARD:

THEOREM 3. *Under the following assumptions:*

1. *The data distributions of individual columns in $\mathcal{G}$ are independent,*
2. *For the value combination $\mathbf{v}$, its occurrence is the result of an independent Bernoulli trial, with the success (occurrence) probability $f_{\mathbf{v}}$, and*
3. *The occurrences of individual possible value combinations are independent,*

the expected number of distinct value combinations is

$$E[D_{\mathcal{G}}] = M - \sum_{\mathbf{v}\in\mathcal{V}} (1 - f_{\mathbf{v}})^N. \quad (2)$$

Proof Sketch For a value combination $\mathbf{v} \in \mathcal{V}$, let a random variate $X_{\mathbf{v}}$ = 1 if the combination appears in R, and 0 otherwise. Then $D_{\mathcal{G}} = \sum_{\mathbf{v}\in\mathcal{V}} X_{\mathbf{v}}$. If we think of each tuple in R as independently drawn from M possible value combinations, then the probability of value combination $\mathbf{v}$ not chosen in any particular draw is $(1 - f_{\mathbf{v}})$. Therefore, under assumption 2, the probability of the combination $\mathbf{v}$ not chosen in any of the N draws and thus not appearing in R is $\Pr(X_{\mathbf{v}} = 0) = (1 - f_{\mathbf{v}})^N$. Hence the expectation of $X_{\mathbf{v}}$ is $E[X_{\mathbf{v}}] = 1 - (1 - f_{\mathbf{v}})^N$. Since $X_{\mathbf{v}}$'s are independent according to assumption 3, $E[D_{\mathcal{G}}] = \sum_{\mathbf{v}\in\mathcal{V}} E[X_{\mathbf{v}}] = \sum_{\mathbf{v}\in\mathcal{V}} \left(1 - (1 - f_{\mathbf{v}})^N\right) = M - \sum_{\mathbf{v}\in\mathcal{V}} (1 - f_{\mathbf{v}})^N$. □

Applying sanity bounds, we use $\widehat{D}_{\text{mar}} = \min\{E[D_{\mathcal{G}}], N\}$ as the estimate. (The subscript $\mathcal{G}$ is dropped when there is no ambiguity.) Also, in the particular case of two attributes, we can use the tighter upper/lower bounds to bound the estimate.

EXAMPLE 3. *Now we revisit the problem of estimating the number of value combinations $D_{\mathcal{G}}$ according to Theorem 3 under the settings of Example 1. It turns out that when $N = 100$, $D_{\mathcal{G}} = 5.94$; while when $N = 11$, $D_{\mathcal{G}} = 3.24$. These results are remarkably different from the estimates suggested in Example 1.*

It is worth pointing out that assumption 3 in Theorem 3 is actually not true in theory, because the frequency vectors $\mathbf{f}_1, \ldots, \mathbf{f}_m$ actually impose constraints on the occurrences of different combinations . For example, if we know $f_{11} \cdot N = 2$, then *exactly* two value combinations with $A_1 = v_{11}$ must appear in the table (where v_{11} is the corresponding value of f_{11}). Therefore, the occurrences of value combinations with $A_1 = v_{11}$ are *not* independent. Nonetheless, in practice, when the number of the possible combinations and the number of tuples are both large, individual constraints do not have a heavy impact on the occurrence frequencies, and assumption 3 approximately holds. This is confirmed by our experimental results.

4.4 Computing the estimate

To estimate COLSCARD, we need to first obtain $\mathbf{f}$, and then compute the estimate using Equation 2, as illustrated in Algorithm 2.

Algorithm 2 Computing the estimate

```
1: /* compute f_G = f_1 ⊗ f_2 ⊗ ... ⊗ f_m */
2: f_G = f_1;
3: for i = 2 to m do
4:   f^old = f_G;
5:   f_G = {}; /* empty vector */
6:   for j = 1 to |f^old| do
7:     for k = 1 to d_i do
8:       f_G = concat(f_G, f_j^old × f_ik); /* append a component to
         the vector f*/
9:     end for
10:  end for
11: end for
12: /* compute the expected number of distinct value combinations
    */
13: S = 0;
14: for j = 1 to M do
15:   S = S + (1 − f_i)^N;
16: end for
17: D̂_mar = M − S;
```

When m or d_i's are large, the algorithm may take a considerable amount of time to run. Note that most of the time is spent on the computation of the Kronecker product. Therefore, we propose to use a sampling approach to reduce the cost. That is, we compute only a random selected portion of the $f_{\mathbf{v}}$'s, and obtain $\widehat{D}_{\text{mar}}$ with proper scaling. More specifically, if we are to draw a sample of size M_s, then at each step, we randomly draw an element from each $\mathbf{f}_i$, and include their product into the sample s, until the size of the sample s reaches M_s. The sample estimate $\widehat{D}^s_{\text{mar}}$ is computed according to Eq. 2. The final estimate $\widehat{D}_{\text{mar}}$ is then obtained by scale $\widehat{D}^s_{\text{mar}}$ up by the factor $\frac{M}{M_s}$. Using this sampling approach, only M_s frequencies are computed as opposed to M in the exact version; therefore the computation time can be reduced by a factor of $\frac{M}{M_s}$. In Section 6, we will show by experiments that this sampling scheme does work well, especially when the frequencies are relatively uniform.

4.5 Optimal histograms for estimating the number of distinct value combinations

Although knowing the exact marginals for each involved attribute can help us obtain accurate estimates if the assumptions hold, we often do not have such luxury. It is common practice in database systems that only concise synopses are maintained for the under-

lying data because of practical considerations such as storage and computational efficiency. The synopses are used to reconstruct an approximate distribution of the data for various purposes, e.g., selectivity estimation, and approximate query answering. A number of choices exist for such synopses [1]; probably the most popular one is histograms [15], which have been extensively used in commercial database systems. Naturally, we would like to consider how to obtain the optimal histograms for our purpose of estimating the number of distinct value combinations. Two broad classes of histograms, namely partition-based histograms, and end-biased histograms, are considered here. Partition-based histograms refer to the histograms that are obtained by partitioning the value domain (with values sorted according to some sort parameter, following [15]) into buckets and approximating the frequencies in each bucket by a succinct measure (most commonly the average). Examples of histograms belonging to this class include the widely used equi-depth and equi-width histograms. End-biased histograms, on the other hand, are obtained by first picking a number of values based on certain criteria, storing the frequencies of those values exactly, and then approximating the frequencies of the remaining values by a succinct measure (say, the average).

We consider the following question for each class of histograms: Given the frequency vectors (either exact or approximate) of all other attributes in $\mathcal{G}$, what is the optimal histogram for attribute A_k for estimating COLSCARD? Let $\mathbf{f}^{\bar{k}}$ be the Kronecker product of the frequency vectors excluding that of attribute A_k, i.e, $\mathbf{f}^{\bar{k}} = \mathbf{f}_1 \otimes \ldots \otimes \mathbf{f}_{k-1} \otimes \mathbf{f}_{k+1} \ldots \otimes \mathbf{f}_m$. Note that those frequency vectors can be either exact or obtained through reconstruction from histograms etc.. For a given amount (B) of buckets, the optimal histogram within each class is defined as the one that achieves an estimate $\widehat{D}_{\text{his}}$ that is closest to $\widehat{D}_{\text{mar}}$, i.e., it minimizes the following error measure:

$$\begin{aligned} ERR_{\text{abs}} =& |\widehat{D}_{\text{his}} - \widehat{D}_{\text{mar}}| \\ =& |(M - \sum_{j=1}^{d_k}(1 - \mathbf{f}^{\bar{k}}\hat{f_{kj}})^N) - (M - \sum_{j=1}^{d_k}(1 - \mathbf{f}^{\bar{k}}f_{kj})^N)| \end{aligned} \quad (3)$$

where $\hat{f_{kj}}$ is the approximate frequency equal to the average of all frequencies in that bucket. We take a little liberty with the mathematical notations in the above equation: we have written $\sum_i \sum_{j=1}^{d_k}(1 - f_i^{\bar{k}}\hat{f_{kj}})^N$ as $\sum_{j=1}^{d_k}(1 - \mathbf{f}^{\bar{k}}\hat{f_{kj}})^N$ for the sake of brevity. Note that since the function $h(x) = (1 - \mathbf{f}^{\bar{k}}x)^N$ is convex, $\widehat{D}_{\text{his}} \geq \widehat{D}_{\text{mar}}$. Therefore the expression for ERR_{abs} can be simplified as

$$ERR_{\text{abs}} = \sum_{j=1}^{d_k}(1 - \mathbf{f}^{\bar{k}}f_{kj})^N - \sum_{j=1}^{d_k}(1 - \mathbf{f}^{\bar{k}}\hat{f_{kj}})^N$$

Partition-based histograms

The error incurred by the approximation made for the frequency in an interval $[a, b]$ is

$$ERR([a,b]) = \sum_{j=a}^{b}(1 - \mathbf{f}^{\bar{k}}f_{kj})^N - (1 - \mathbf{f}^{\bar{k}}\hat{f}_{[a,b]})^N$$

where $\hat{f}_{[a,b]} = \frac{\sum_{j=a}^{b} f_{kj}}{b-a+1}$. Let $ERR^*(i,l)$ be the minimum ERR for the frequency vector $(v_{k1}, \ldots, v_{ki})$ using at most l buckets. Then we have the following observation:

$$ERR^*([i,l]) = \min_{1 \leq t < i}\{ERR^*(t, l-1) + ERR([t+1, i])\},$$

$\mathbf{f}_k$	Index of the frequency stored			
	1	2	3	4
$(.01, .02, .03, .94)$	1.6818	2.0067	2.1788	**.1526**
$(.01, .29, .31, .39)$	**.0097**	1.0978	1.0883	1.0208

(a) $B = 2$

$\mathbf{f}_k$	Indexes of the frequencies stored					
	[1,2]	[1,3]	[1,4]	[2,3]	[2,4]	[3,4]
$(.01, .02, .03, .94)$	.7914	.9666	**.0239**	1.2944	.1526	.0559
$(.01, .29, .31, .39)$	**.0049**	.0085	.0005	1.0632	.9540	.9212
$(.01, .02, .10, .87)$	.3524	.9602	.2439	1.2876	0.5112	**.0559**

(b) $B = 3$

Figure 1: Errors of end-biased histograms

which means that finding the optimal histogram of l buckets can be reduced to the case of finding an optimal histogram of $l-1$ buckets. This leads to a dynamic programming algorithm similar to that proposed for V-optimal histograms [10]. Detailed algorithmic development is omitted here because of space limitations.

End-biased histograms

An end-biased histogram with B-buckets stores the exact frequencies of $B-1$ attribute values in $B-1$ singleton buckets, and the average of the frequencies of the remaining values in one "flat" bucket. The most widely used type of end-biased histograms keeps the exact frequencies of the most frequent values in hopes of capturing the majority "mass" of the frequency distribution. For our purpose of estimating COLSCARD, however, cropping the largest frequencies is no longer always the best option.

EXAMPLE 4. *Suppose* $\mathbf{f}^{\bar{k}} = [0.1, 0.9]$*, N=10. Consider the optimal end-biased histograms in the following two cases: (i)* $\mathbf{f}_k = (0.01, 0.02, 0.03, 0.94)$*, and (ii)* $\mathbf{f}_k = (0.01, 0.29, 0.31, 0.39)$*. Figure 1(a) shows the* ERR_{abs} *associated with each choice of frequency when there is only one frequency (i.e.,* $B = 2$*) to be stored exactly. Observe that for case (i), choosing the largest frequency (0.94) results in the smallest error among all choices. For case (ii), however, the optimal choice is the smallest frequency (0.01).*

What if more than one frequency is stored exactly? Figure 1(b) shows the results for the case of $B = 3$*; i.e., two frequencies are chosen to be stored exactly. For illustration purpose, we also consider a third case, (iii)* $\mathbf{f}_k = (0.01, 0.02, 01, 0, 87)$*. For case (i), storing the combination of the smallest and the largest frequencies is optimal, while for case(ii), it is the best to store the two smallest frequencies. For case (iii), on the other hand, storing the two largest frequencies leads to the smallest error.*

A naive approach for identifying the optimal end-biased histogram for attribute A_k would be to consider all possible combinations of $B-1$ elements from the frequency vector $\mathbf{f}_k$. The errors incurred by storing each of the combinations exactly are computed and the one with the smallest error is chosen. This algorithm takes time proportional to $\binom{d_k}{B-1}$, and thus scales poorly with respect to d_k.

Fortunately, the optimal choice of the set of frequencies takes a special form, which can be leveraged to safely ignore a large portion of the possible combinations. A rather counter-intuitive and surprising result, as we will show, is that optimality is achieved as the smallest, the largest, or a combination of the smallest and the largest frequencies are chosen to be stored exactly.

THEOREM 4. *Without loss of generality, assume that the frequencies of the values in attribute A_k are sorted such that $f_{k1} \leq f_{k2} \leq \ldots \leq f_{kd_k}$. Let $ERR_{abs}(\mathcal{S})$ be the error when the set of frequencies $\mathcal{S}$ is chosen to be stored exactly. With a space budget of B buckets, the set of frequencies to be stored exactly in an optimal end-biased histogram takes the form of $\mathcal{S}(l) = \{f_{k1}, \ldots, f_{kl}\} \cup \{f_{ku}, \ldots, f_{kd_k}\}$, $(u-l = d_k - B+2)$, and there exists an optimal l such that $ERR_{abs}(\mathcal{S}(l))$ is minimized.*

Proof Sketch We first prove that the optimal set of frequencies to be stored exactly takes the form of $\{f_{k1}, \ldots, f_{kl}\} \cup \{f_{ku}, \ldots, f_{kd_k}\}$, $(u - l = d_k - B + 2)$; i.e., they are chosen from the smallest and largest frequencies.

To show this, we examine the monotonicity of the error function,

$$ERR_{\text{abs}} = T - \sum_{j:f_{kj} \in \mathcal{F}_E} (1 - \mathbf{f}^{\bar{k}} f_{kj})^N - (M - B + 1) \left[1 - \mathbf{f}^{\bar{k}} \frac{1 - \sum_{j:f_{kj} \in \mathcal{F}_E} f_{kj}}{M - B + 1}\right]^N,$$

where $T = \sum_{j=1}^{d_k} (1 - \mathbf{f}^{\bar{k}} f_{kj})^N$, and $\mathcal{F}_E$ is the set of $B-1$ frequencies to be stored exactly. We have

$$\frac{\partial ERR_{\text{abs}}}{\partial f_{kj}} = N\mathbf{f}^{\bar{k}}(1 - \mathbf{f}^{\bar{k}} f_{kj})^{N-1} - N\mathbf{f}^{\bar{k}} \left[1 - \mathbf{f}^{\bar{k}} \frac{1 - \sum_{j:f_{kj} \in \mathcal{F}_E} f_{kj}}{M - B + 1}\right]^{N-1},$$

and

$$\frac{\partial^2 ERR_{\text{abs}}}{\partial f_{kj}^2} = -N(N-1)\mathbf{f}^{\bar{k}2}(1 - \mathbf{f}^{\bar{k}} f_{kj})^{N-2} - N(N-1) \cdot \mathbf{f}^{\bar{k}2} \frac{1}{M - B + 1} \left[1 - \mathbf{f}^{\bar{k}} \frac{1 - \sum_{j:f_{kj} \in \mathcal{F}_E} f_{kj}}{M - B + 1}\right]^{N-2}.$$

Since $0 < f_{kj} \leq 1$, we have $\frac{\partial^2 ERR_{\text{abs}}}{\partial f_{kj}^2} \leq 0$, which implies that the minimum of ERR_{abs} is reached at either end of the range of available f_{kj} values.

Suppose there exists an optimal set of frequencies taking the form of $\{f_{k1}, \ldots, f_{kl}\} \cup \{f_{k_\alpha}, \ldots, f_{k_\beta}\} \cup \{f_{ku}, \ldots, f_{kd_k}\}$ $(u - l + \alpha - \beta = d_k - B + 3)$ such that $\alpha - l > 1$ and $u - \beta > 1$, i.e., there are frequencies not belonging to either the set of smallest frequencies or the set of largest frequencies. Because of the concavity of ERR_{abs}, it is trivial to show that replacing any of the frequencies in $\{f_{k_\alpha}, \ldots, f_{k_\beta}\}$ by either $f_{k(l+1)}$ or $f_{k(u-1)}$ will reduce ERR_{abs}. Therefore, the optimal set of frequencies cannot contain any frequency not belonging to the contiguous group of either the smallest or the largest frequencies. Hence, we conclude that the optimal set of frequencies to be stored exactly must take the form of $\{f_{k1}, \ldots, f_{kl}\} \cup \{f_{ku}, \ldots, f_{kd_k}\} (u - l = d_k - B + 2)$.

To obtain the optimal set of frequencies, we simply need to identify the optimal l (or u) that minimizes ERR_{abs},

$$\min_{0 \leq l \leq d_k} \{\{f_{k1}, \ldots, f_{kl}\} \cup \{f_{ku}, \ldots, f_{kd_k}\}\} (u - l = d_k - B + 2). \square$$

Theorem 4 makes efficient computation of end-biased histograms possible. To obtain the optimal end-biased histogram, one simply needs to evaluate $ERR_{\text{abs}}(\mathcal{S}(l))$ for $0 \leq l \leq B-1$, and choose the l corresponding to the smallest $ERR_{\text{abs}}(\mathcal{S}(l))$. Clearly, the cost of such computation is linear in B, and is *independent* of d_k.

5. EXPERIMENTAL RESULTS

In this section, we report the results of an experimental evaluation of the proposed estimation methods.

5.1 Data sets

We studied the accuracy of the proposed estimator on synthetic as well as two real data sets "Cover Type" and "Census Income" downloaded from UCI Machine Learning repository[16] previously used in the literature for evaluating estimators of the number of distinct values [4, 3].

Synthetic data were used to study the property of the proposed estimator in a controlled manner. Specifically, the synthetic data were generated by varying the following characteristics:

- Data skew: The data were generated from Zipfian distribution with parameter z ranging from 0 (uniform distribution) to 4 (very highly-skewed distribution). We put the data generated from different Z values into different attributes of the same table. Therefore, each attribute has a different degree of skew.
- Number of tuples: We generated tables of different sizes, with N ranging from 10K to 1M.

The first real data set "Cover Type" is the Forest Covertype data from the National Forest Service. It has 581,012 tuples and 10 quantitative attributes, with the number of distinct values in individual attributes ranging from 67 to 5,827. The second real data set "Census Income" is a data set derived from the U.S. Census database. It has 32,561 tuples and 14 attributes, with the number of distinct values in individual attributes ranging from 2 to 21,648.

5.2 Error measure

The common error metric used in existing literature [4, 3] is the em ratio error metric, which is defined as $error = \max\{\frac{D}{\hat{D}}, \frac{\hat{D}}{D}\}$, the ratio of the estimator and the true COLSCARD, where the ratio is inverted when necessary to ensure that the error is always greater than 1. We find it more intuitive to use a slightly modified error metric $ERR = error - 1$ because ERR is 0 when the estimate coincides with the true value. Therefore, ERR is used in our evaluation, and oftentimes, we use percentage representation in the text because it is more revealing.

5.3 Results on synthetic data

Effect of data skew Figure 2 shows the effect of data skew on the proposed estimator as well as the naive estimator. We consider attribute pairs with different degrees of skewness, using z_1 and z_2 to denote the Zipfian parameter of the two attributes respectively. The number of tuples is 100K. The first pair of attributes are both uniformly distributed ($z_1 = 0, z_2 = 0$), with the number of distinct values in both attributes being 1K. Because of the uniformity and the large possible number of combinations ($1K \times 1K = 1M$), nearly every tuple is different. In this case, the naive estimator, which gives an estimate of 100K, does pretty well, though not as good as our proposed estimator, which produces an error of less than 0.1% and is thus almost invisible in the plot. When data become skewed, the naive estimator deteriorates very fast, giving errors in excess of 500%. The proposed estimator is highly accurate when one of the attributes involved is uniformly distributed, though the error slightly increases when the skewness in one of the attributes grows. The estimator gives larger errors for the case where both attributes are highly skewed ($z_1 = 4, z_2 = 4$). However, in all cases, the proposed estimator produces small errors, demonstrating its robustness to data skew. Since the proposed estimator is orders of magnitudes more accurate than the naive estimator, we do not include the naive estimator in the following experiments.

Effect of number of tuples Our next experiment studies the effect of the number of tuples on the estimation accuracy (Figure 3).

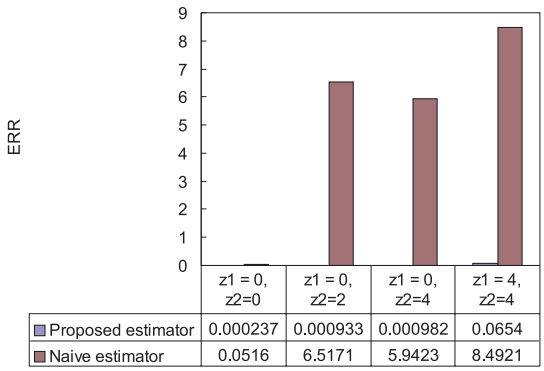

Figure 2: Accuracy vs. data skew

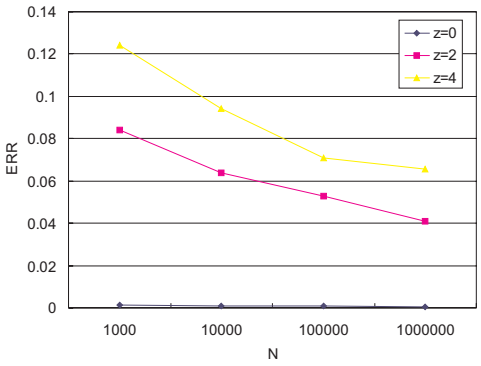

Figure 3: Accuracy vs. number of tuples

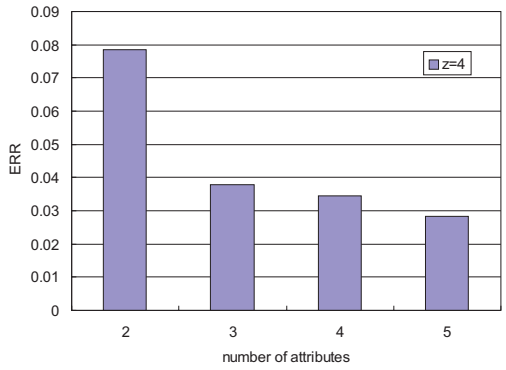

Figure 4: Accuracy vs. number of attributes

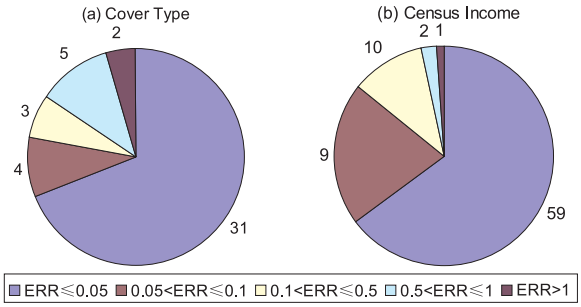

Figure 5: Errors on the two real data sets

We use three synthetic data sets with two attributes. In each data set, the two attributes follow the Zipfian distribution with the same parameter z. An interesting observation is that as N increases, the estimation error actually decreases. This can be partly explained by the fact that the "randomness" in the number of distinct value combinations appearing in the table decreases as N goes up. In the extreme case, when N approaches infinity, every possible combination is expected to appear in the table, and COLSCARD therefore has a very low degree of randomness.

Effect of number of attributes We now study the effect of the number of attributes on the estimation accuracy. We estimate the COLSCARDs for groups of 2, 3, 4, 5 attributes respectively. Figure 4 shows a typical result on the data set with $z = 4$ for all attributes. It is worth noting that the accuracy improves as the number of attributes increases, which indicates that the proposed estimator can handle large groups of attributes as well. Therefore, for other experiments, we only present the results for the case of two attributes.

5.4 Results on real data

We carried out experiments on the two real data sets, and the results are reported below.

Error profile For both data sets, we experimented with all possible attribute pairs (45 pairs for the Cover Type data, and 91 pairs for Census Income data), and the results are summarized in Figure 5 showing the percentage of errors falling in each error range. As can be seen from Figure 5, the majority of the pairs have an estimation error of less than 5%. We looked closer into the cases where the error is greater than 100%, and discovered that those pairs are all strongly correlated. For example, the pair in the Census Income data set with an error greater than 100% consists of "education"(Bachelor's, Prof-school, etc.) and "education-num" (number of years of eduction), which are clearly very strongly correlated; therefore, the value combinations appearing in the data set is small (with COLSCARD being 32 while each attribute has 16 distinct values). Since our estimator assumes that the attributes are independent, it significantly overestimates the COLSCARD. If we have no knowledge available about the correlation, there is not much we can do to address this problem. However, when there exists information about correlation, it is possible to obtain better estimates. We plan to look into such cases for future work.

Sample rate In Section 3, we discussed using sampling to reduce the computational cost. The next experiment studies the effect of sample rate on the accuracy. In Figure 6, we show a typical result, obtained from estimating the COLSCARD for the first two attributes in the Cover Type data set (namely, "Elevation" and "Aspect"). As shown in the plot, the accuracy improves as sample rate increases. A similar trend was observed on other attribute combinations.

End-biased histograms We showed how to obtain an optimal end-biased histogram in Section 4. We conducted experiments to evaluate the accuracy of the histograms. Figure 7 shows the estimation error due to the use of the end-biased histograms on the "capital gain" attribute (with 119 distinct values) in the attribute pair of "work class" and "capital gain", as the number of buckets varies. As expected, the accuracy improves as the number of buckets increases. Similar results are obtained on other attributes.

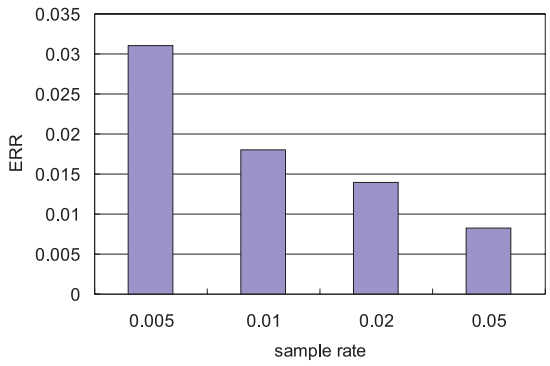

Figure 6: Accuracy vs. sample rate

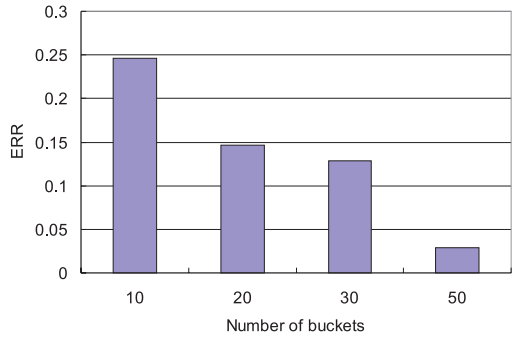

Figure 7: Accuracy of end-biased histograms

6. CONCLUSIONS AND FUTURE WORK

Estimating the number of distinct value combinations is an important task for both query optimization and approximate query answering. Current commercial systems often have to settle with naive estimates (with large errors) in the absence of important statistical information on the attribute group involved. Different from previous work which studies the estimation of the number of distinct values in a single attribute, we attack the more general problem of estimation for multiple attributes. Instead of extending the sampling framework that has been extensively used for estimation for single attributes, we take a complimentary approach by focusing on utilizing existing knowledge maintained in the database catalog. We discussed the case where exact frequency information is available on individual attributes, and derived a pair of upper/lower bounds on the COLSCARD for attribute pairs. We also proposed an estimator by treating the frequency information as probabilities and computing the expected number of distinct combinations appearing in the table. For cases where only partial information is available, we studied how histograms can be used to get an approximate estimate, and we show optimality results for two widely used types of histograms. Experimental results have shown the effectiveness of the proposed estimator.

For future work, we would like to consider the case where information is available on some subsets of the attribute group for which an estimate is required. Examples of such information include joint indexes, multidimensional histograms, statistical views, etc. They offer the opportunity of improving the estimation accuracy, because they usually capture some correlation information between attributes. A key challenge here is to make consistent usage of all sources of information. The iterative scaling method, which was previously used for selectivity estimation of conjuncts of predicates[11], can be helpful in this regard.

7. REFERENCES

[1] D. Barbará, W. DuMouchel, C. Faloutsos, P. J. Haas, J. M. Hellerstein, Y. E. Ioannidis, H. V. Jagadish, T. Johnson, R. T. Ng, V. Poosala, K. A. Ross, and K. C. Sevcik. The New Jersey data reduction report. *IEEE Data Eng. Bull.*, 20(4):3–45, 1997.

[2] J. Bunge and M. Fitzpatrick. Estimating the number of species: A review. *Journal of the American Statistical Association*, 88:364–373, 1993.

[3] M. Charikar, S. Chaudhuri, R. Motwani, and V. Narasayya. Towards estimation error guarantees for distinct values. In *PODS*, pages 268–279, 2000.

[4] P. B. Gibbons. Distinct sampling for highly-accurate answers to distinct values queries and event reports. In *VLDB*, pages 541–550, 2001.

[5] P. J. Haas, J. F. Naughton, S. Seshadri, and L. Stokes. Sampling-based estimation of the number of distinct values of an attribute. In *VLDB*, pages 311–322, 1995.

[6] P. J. Haas and L. Stokes. Estimating the number of classes in a finite population. *Journal of the American Statistical Association*, 93:1475–1487, 1998.

[7] J. M. Hellerstein, P. J. Haas, and H. J. Wang. Online aggregation. In *SIGMOD Conference*, pages 171–182, 1997.

[8] W.-C. Hou, G. Özsoyoglu, and B. K. Taneja. Statistical estimators for relational algebra expressions. In *PODS*, pages 276–287, 1988.

[9] W.-C. Hou, G. Özsoyoglu, and B. K. Taneja. Processing aggregate relational queries with hard time constraints. In *SIGMOD Conference*, pages 68–77, 1989.

[10] H. V. Jagadish, N. Koudas, S. Muthukrishnan, V. Poosala, K. C. Sevcik, and T. Suel. Optimal histograms with quality guarantees. In *VLDB*, pages 275–286, 1998.

[11] V. Markl, N. Megiddo, M. Kutsch, T. M. Tran, P. Haas, and U. Srivastava. Consistently estimating the selectivity of conjuncts of predicates. In *VLDB*, 2005.

[12] J. F. Naughton and S. Seshadri. On estimating the size of projections. In *ICDT*, pages 499–513, 1990.

[13] F. Olken and D. Rotem. Random sampling from databases - a survey, March 1995.

[14] G. Özsoyoglu, K. Du, A. Tjahjana, W.-C. Hou, and D. Y. Rowland. On estimating COUNT, SUM, and AVERAGE. In *DEXA*, pages 406–412, 1991.

[15] V. Poosala, Y. E. Ioannidis, P. J. Haas, and E. J. Shekita. Improved histograms for selectivity estimation of range predicates. In *SIGMOD Conference*, pages 294–305, 1996.

[16] C. B. S. Hettich and C. Merz. UCI repository of machine learning databases, 1998.

Trademarks