Estimating the Confidence of Conditional Functional Dependencies

Graham Cormode
AT&T Labs–Research
graham@research.att.com

Lukasz Golab
AT&T Labs–Research
lgolab@research.att.com

Flip Korn
AT&T Labs–Research
flip@research.att.com

Andrew McGregor
University of Massachusetts Amherst
mcgregor@cs.umass.edu

Divesh Srivastava
AT&T Labs–Research
divesh@research.att.com

Xi Zhang
SUNY Buffalo
xizhang@cse.buffalo.edu

ABSTRACT

Conditional functional dependencies (CFDs) have recently been proposed as extensions of classical functional dependencies that apply to a certain subset of the relation, as specified by a *pattern tableau*. Calculating the support and confidence of a CFD (i.e., the size of the applicable subset and the extent to which it satisfies the CFD) gives valuable information about data semantics and data quality. While computing the support is easier, computing the confidence exactly is expensive if the relation is large, and estimating it from a random sample of the relation is unreliable unless the sample is large.

We study how to efficiently estimate the confidence of a CFD with a small number of passes (one or two) over the input using small space. Our solutions are based on a variety of sampling and sketching techniques, and apply when the pattern tableau is known in advance, and also the harder case when this is given after the data have been seen. We analyze our algorithms, and show that they can guarantee a small additive error; we also show that relative errors guarantees are not possible. We demonstrate the power of these methods empirically, with a detailed study using both real and synthetic data. These experiments show that it is possible to estimate the CFD confidence very accurately with summaries which are much smaller than the size of the data they represent.

Categories and Subject Descriptors

H.2.7 [**Database Management**]: Database Administration

General Terms

Algorithms, Management, Theory

Keywords

conditional functional dependencies

Table 1: Tableau for $\{\texttt{age\_grp}, \texttt{education}, \texttt{occupation}\} \rightarrow \texttt{salary\_grp}$ on census data

age_grp	education	occupation	salary_grp
__	Masters	Schoolteacher	__
__	PhD	Professor	__

1. INTRODUCTION

Conditional functional dependencies (CFDs) have recently been proposed to characterize the semantics of complex data and facilitate data cleaning [3, 4, 10, 11, 14, 15, 16, 18]. A CFD is a functional dependency (FD) that holds on a subset of the relation specified in an accompanying *pattern tableau*. The *support* of a CFD is the fraction of rows that match the tableau. Its *confidence*, defined formally below, is the fraction of rows satisfying the functional dependency amongst those that match the tableau. The support and confidence of a given tableau are key properties in understanding and exploring data quality. Consider a data warehouse containing large quantities of historical data. Careful analysis of a particular relation over time may have revealed various tableaux that have high support and confidence; checking the support and confidence on new data as they arrive hour by hour is an important quality check, and can reveal new trends if the support or confidence change suddenly. A second situation is when analysis has revealed a new tableau that has high support and confidence on the new data; it is then natural to ask over which historical data this tableau also had high support and confidence. Because of the quantity of the data, and the many different tableaux which may be considered, we seek efficient methods to find the support and confidence of CFDs in large relations.

In these settings, as advocated in [8], although the full data may be stored exactly, it is desirable to also maintain various *compact summaries* as new data and/or new sources are integrated over time. Summaries enable rapid data auditing without having to run complex queries on massive fact tables, and when some data are physically stored at remote locations or archived on slow storage devices. We seek compact summaries for computing the support and confidence of CFDs in order to understand changes in data semantics. Ideally, these summaries should be easy to derive from the base data. They should require only one or a few passes over the data to compute, so that they can be computed as the data is loaded into a warehouse, without additional costly sorting steps. To make this precise, we describe CFDs in more detail, and go on to define the properties we desire from compact summaries for support and confidence estimation.

SIGMOD'09, June 29–July 2, 2009, Providence, Rhode Island, USA.
Copyright 2009 ACM 978-1-60558-551-2/09/06 ...$5.00.

Table 2: Tableau for $\{\texttt{supplier}, \texttt{part}, \texttt{part\_type}\} \rightarrow \texttt{price}$

supplier	part	part_type	price
__	__	tire	__
U.S. Auto Parts	__	__	__
__	325_headlight	__	50

Consider the FD $\{\texttt{age\_grp}, \texttt{education}, \texttt{occupation}\} \rightarrow \texttt{salary\_grp}$ on a census table, which states that individuals in the same age group that have the same education and occupation must have salaries in the same group. This rule is not expected to hold over the entire database; e.g., accountants working in different companies may earn different salaries despite having the same age group and education. It may, however, be true for certain government-regulated occupations with age-based salary scales, e.g., high school teachers with Masters' degrees and professors with Doctorate degrees. Table 1 illustrates the corresponding tableau. This indicates that the FD should hold on the set of tuples that match at least one of the listed patterns. The wildcard pattern "__" matches all possible attribute values.

In addition to limiting the scope of an FD, a tableau may specify tuple-level constraints. Consider a data warehouse of orders placed by an auto manufacturer to its suppliers. Table 2 illustrates a tableau for the FD $\{\texttt{supplier}, \texttt{part}, \texttt{part\_type}\} \rightarrow \texttt{price}$, i.e., a supplier charges a constant price for a part (of some type `part_type`). According to the first two patterns, this is true for tires and parts supplied by U.S. Auto Parts, respectively. According to the third pattern, a supplier of headlights for model 325 must charge a constant price, but not any constant price; it must be 50. Thus, constants in the left-hand-side attributes define the scope of the CFD and constants on the right-hand-side restrict attribute values. Note that patterns may overlap as U.S. Auto Parts may supply tires and/or 325_headlights.

Exceptions can occur in real data in the form of incorrect records (e.g., a supplier charges the wrong price of 325_headlights) or correct records that deviate from the assumed semantics (e.g., some professors in a given age group may earn lower or higher salaries than normal). Thus, even *conditional* functional dependencies may not hold exactly. Instead, we consider approximate CFDs. Following prior work [18, 19, 23], we define the *confidence* of a CFD as the maximum fraction of tuples that may be retained, so that if all other tuples were deleted, the remaining ones would satisfy the CFD with no exceptions. The *support* is the fraction of tuples that match at least one pattern in the tableau. These quantities are fundamental to using CFDs to analyze data quality, and perform data auditing, and so our focus in this work is on problems related to estimating the confidence in various settings.

Given a supplied FD, one type of audit estimates the support and confidence of an existing tableau on new data. Another audit involves computing the support and confidence of new tableaux on historical data. For instance, the tableau from Table 1 may not hold on the latest census data due to changes in the job market. Instead, the tableau (__|Bachelors|Nurse||__) may be identified, in which case a natural question is to study how the support and confidence of this tableau has varied over time, and to identify when it first became significant. Similarly, the tableau from Table 2 may not hold on new data, perhaps because the contract with U.S. Auto Parts has been re-negotiated to allow variable prices, or because some prices are incorrectly recorded. These analyses may be time-consuming to compute exactly over archived data (and impossible if the data is not stored in full); instead, we look for more practical ways to answer these questions from pre-computed summaries.

For many query types, summarizing data with a random sample and evaluating the query over the sample is a good way to approximate the answer. However, while small random samples can give good estimates of simple frequency counts, such as the support of a CFD, estimating the confidence of a CFD is more complex. In fact, a uniform random sample of size $\Omega(\frac{\sqrt{N}}{\epsilon})$ is required even to estimate whether a standard FD holds on a relation of size N with confidence above $1 - \epsilon$ [23]. Thus we need a different approach to building summaries for CFD confidence estimation.

1.1 Our Contributions

In this paper, we provide the first scalable summaries (whose space complexity is independent of N) that estimate the true confidence of a CFD (not just an indication of whether it is above or below some fixed threshold) for effective data auditing. Our contributions include:

- We prove lower bounds, in terms of relative and absolute errors, for confidence estimation by any algorithm that makes a small number of passes over the input.
- We propose a two-pass algorithm with space complexity $O(\frac{1}{\epsilon^3} \log \frac{1}{\delta})$ that estimates the confidence of a given CFD within additive error ϵ, with probability $1 - \delta$. The summary is based on a random sample of rows made in the first pass, while the second pass computes additional information about the sampled data.
- We propose a second algorithm, inspired by the first, which constructs a summary in a single pass using the same amount of space. This algorithm interleaves reservoir sampling [27] and the gathering of additional information. The modified algorithm does not give guarantees for the worst case, but is shown to work well in practice.
- We propose a more complex one-pass algorithm, based on non-uniform sampling and count-min sketches [13], that achieves the same error bounds and asymptotic space usage as the two-pass algorithm.
- We test our solutions on a variety of real and synthetic data sets, and provide guidelines for choosing the best strategy.

The summaries generated by our algorithms may be used to estimate the confidence of arbitrary CFDs sharing a fixed FD (i.e. the attributes considered are fixed, but the tableau can vary). This allows the confidence of candidate tableaux to be estimated, and accommodates testing new CFDs on summaries of past data. The single-pass algorithms do not require a priori knowledge of the data size, meaning that a summary of newly arrived data may be built on-the-fly, and made available on-demand for real-time auditing.

1.2 Paper Outline

In the remainder of this paper, Section 2 formalizes the problem statement and discusses straightforward approaches to confidence computation. The two-pass algorithm is presented in Section 3, and the two one-pass algorithms in Sections 4 and 5. Section 6 proves lower bounds for confidence estimation in limited time and space. Section 7 experimentally evaluates the proposed algorithms, and related work is discussed in Section 8.

2. PRELIMINARIES

2.1 Definitions

Let S be a relational schema with attributes $A_1, A_2 \ldots A_\ell$. Consider a table R over this schema, with rows $\{r_1, r_2, \ldots r_N\}$.

DEFINITION 1. *A* functional dependency *(FD) $X \rightarrow Y$ is said to hold over sets of attributes X and Y on R if*

$$\forall i, j . r_i[X] = r_j[X] \Rightarrow r_i[Y] = r_j[Y],$$

where $r[X]$ denotes the row tuple r projected on the attributes X.

In other words, $X \rightarrow Y$ over the relation R if the value of X uniquely determines the value of Y. Here, the set $X \subseteq S$ is referred to as the antecedent, and $Y \subseteq S$ is called the consequent.

The requirements for a functional dependency are strong. In many cases, a similar condition holds over only a subset of the data. This subset is identified by a condition on the data values, hence this leads to *conditional functional dependencies*.

DEFINITION 2. *A* conditional functional dependency *(CFD) ϕ on S is a pair $(X \rightarrow Y, T_p)$, where $X \rightarrow Y$ is a standard FD, referred to as the embedded FD; and T_p is a "pattern tableau" that defines over which rows of the table the embedded FD applies. Each entry $t_p \in T_p$ specifies a pattern over $X \cup Y$, so for each attribute in $A \in X \cup Y$, either $t_p[A] = a$, where a is a value in the domain of A, or else $t_p[A] =$ ___ , for the special wildcard symbol* ___ *. A row r_i* satisfies *an entry t_p of tableau T_p for attributes A, denoted by $r_i[A] \asymp t_p[A]$, if either $r_i[A] = t_p[A]$, or else $t_p[A] =$ ___ . The CFD ϕ holds if*

$$\forall i, j, p . r_i[X] = r_j[X] \asymp t_p[X] \Rightarrow r_i[Y] = r_j[Y] \asymp t_p[Y].$$

Equivalently, the CFD states that on the subset of rows matching the antecedent of at least one pattern (the *support set*), the FD $X \rightarrow Y$ holds, and so do any tuple-level constraints on the consequent attributes (e.g., the bottom pattern in Table 2). Thus, a standard FD may be expressed as a CFD with a single all-wildcards tableau pattern that matches every tuple (i.e., the support set is the entire relation). The requirements on the antecedents of the embedded FD are referred to as "conditions", while the requirements on the consequents are referred to as "assertions". More generally, these assertions can be arbitrary constant constraints on the consequent, e.g., requiring that the price of headlights be less than 50. An common case is when there are no assertions (only conditions) in the CFD: we refer to this as the assertion-free case.

In many cases, a CFD will not hold exactly. That is, some assertions may not be satisfied, or there may be some rows in the support set which agree on the antecedent (X), but which disagree on the consequent (Y).

DEFINITION 3. *The confidence, $C^{\phi}(R)$, of a CFD ϕ on relation R is the maximum fraction of its support set that can be retained, so that after deleting all other rows, the remainder of the support set satisfies the embedded FD and all relevant assertions.*

Observe that we can consider each distinct antecedent value belonging to the support set separately. We refer to the set of rows associated with a (fully-instantiated) antecedent x as a *group*, or "the group of x". In order to satisfy the CFD, all rows in each group must have the same consequent value and must satisfy all applicable assertions. To minimize the number of rows that would have to be deleted, for each group, we should retain rows with the most common consequent value that meets all applicable assertions. Thus, the confidence can be computed exactly, given a table and a CFD, at the cost of sorting the whole table to form the groups.

Without loss of generality, we can write the table R as a (multi)set of rows $r_i = (x_i, y_i)$, where $x_i \in X$ is the antecedent, $y_i \in Y$ is the consequent, and all other attributes are dropped. Denote the total number of rows in R as N. Define N_x as $|\{r_i : x_i = x\}|$, the number of rows sharing the antecedent x (the size of the group of x), and $N_{x,y}$ as

$$|\{r_i : x_i = x \wedge y_i = y \wedge \forall t_p \in T_p, t_p[X] \asymp x \Rightarrow t_p[Y] \asymp y\}|,$$

the number of rows with antecedent x and consequent y that satisfy all applicable assertions. Furthermore, denote the support set of CFD ϕ on R as $s^{\phi}(R)$, i.e.,

$$s^{\phi}(R) = \{r_i : \exists t_p \in T_p . r_i[X] \asymp t_p[X]\},$$

and define $\text{supp}^{\phi}(R) = \frac{|s^{\phi}(R)|}{N}$ as the support of ϕ on R. Then

$$C^{\phi}(R) = \sum_{x \in s^{\phi}(R)[X]} \max_y \frac{N_{x,y}}{N} = \sum_{x \in s^{\phi}(R)[X]} \frac{N_x}{N} \max_y \frac{N_{x,y}}{N_x}.$$

The fraction N_x/N can be interpreted as the "support" of x, while the fraction $\max_y N_{x,y}/N_x$ can be thought of as a "confidence" of the group x. We refer to the $\max_y N_{x,y}$ rows for x as the set of *keepers* for the group x, since these are the rows that are retained (kept) in order to build the largest possible relation which satisfies the CFD.

2.2 Problems

Given sufficient computational resources, it is straightforward to compute the confidence of a CFD by first sorting the input on attributes XY. For each group in the support set, the number of matching rows (N_x), and the frequency of the most common consequent that satisfies all applicable assertions ($\max_y N_{x,y}$) can be found, allowing direct computation of the confidence. However, we are interested in the case when the input is very large, and we can only afford to access it via a small number of passes, while retaining only a small amount of summary information. We compare to the cost of exact computation in our experimental study later, and show that the cost is considerably higher than our algorithms. Note that a simple uniform sample of rows is sufficient to estimate the support of any given CFD, so this quantity can be assumed to be known approximately.

The first problem we consider involves estimating the confidence of a single CFD that is specified upfront.

DEFINITION 4 (FIXED CFD PROBLEM). *The fixed tableau CFD confidence estimation problem is, given a relation R, a proposed CFD ϕ, and parameters $0 < \epsilon < 1$ and $0 < \delta < 1$ to return an estimate $\hat{C}^{\phi}(R)$ so that*

$$\Pr\left[|\hat{C}^{\phi}(R) - C^{\phi}(R)| > \epsilon\right] < \delta$$

Note that for this problem, any rows in the input that are not in the support set of the specified CFD can be ignored, since they are not relevant to the computation of the confidence. This is related to the problem of estimating the confidence of an FD on the remaining rows; however, a key difference is the presence of assertions in tableau, making this problem more general.

The second problem we consider is to estimate the confidence of CFDs where the embedded FD is fixed but the tableau is provided after the data have been seen. This problem is more general since the entire relation needs to be summarized to accommodate CFDs with various support sets. Therefore, we must relax the desired error bounds, especially for CFDs with small support sets for which only a fraction of the summary will be relevant. The next definition allows the error to scale naturally with the diminished support.

DEFINITION 5 (VARIABLE CFD PROBLEM). *The variable tableau CFD confidence estimation problem is to create a summary based on a relation R, an embedded FD $X \rightarrow Y$, and parameters*

$0 < \epsilon < 1$ *and* $0 < \delta < 1$ *so that given an arbitrary CFD* ϕ *constructed from this embedded FD, we can return an estimate* $\hat{C}^{\phi}(R)$ *with*

$$\Pr\left[|\hat{C}^{\phi}(R) - C^{\phi}(R)| > \frac{\epsilon}{\text{supp}^{\phi}(R)}\right] < \delta$$

2.3 Drawbacks of Existing Approaches

It is natural to ask why simple sampling schemes are not sufficient for the two problems defined above. We show cases where such schemes fail even in the simplest case, in estimating the confidence of the fixed CFD which matches all tuples, __ → __. We show this over a relation R that has one group of size $N/2$, and either this has 1 unique consequent, or $N/2$ unique consequents. The rest of the relation consists of $N/4$ groups each of size 2, and either each group has a unique consequent, or else has two different consequents.

Uniform Row Sampling. Consider a scheme which samples a set of rows uniformly from the relation, and then tries to use this information to estimate the confidence. Let the large group in R have a unique consequent, so the confidence is either 1 or 0.75, depending on how the small groups are arranged. The sampling scheme is unlikely to pick two rows from the same small group, and so has no information to distinguish the two cases, unless the sample size is $\Omega(\sqrt{N})$. □

Uniform Group Sampling. Consider a scheme that samples uniformly from the groups (so each group is equally likely to be picked), and collects some information about the sampled groups. Let the small groups have a unique consequent, so the confidence is either 1 or $0.5 + 2/N$, depending on the consequents in the large group. However, unless some information is collected about the large group, then there is no way to distinguish the two cases. This only occurs if the sample size is $\Omega(N)$ when groups are sampled uniformly. □

These examples show that simple uniform sampling approaches alone will not suffice, and instead we will need to consider more nuanced sampling schemes.

3. TWO-PASS SOLUTION

In this section, we describe a solution to the CFD confidence estimation problem which takes two passes through the data. In the first pass, it samples tuples uniformly from the relation. In the second pass, for each sampled tuple, the confidence of the corresponding group is estimated, then these are used to build an estimator for the overall confidence. We show that this leads to an estimator with strong guarantees.

3.1 Confidence of a Single Group

We first show how to solve a subproblem, to estimate the confidence associated with a single group, denoted by C_x. Equivalently, this is the confidence of a tableau with the pattern $t_p[X] = x$ for the completely specified, or "fully-instantiated" antecedent x. For now, assume that the CFD is assertion-free. We show two ways to build an estimator $\hat{C}_x$ which accurately approximates C_x with a single pass over the input.

3.1.1 Sampling Approach

Given x, we can draw a sample $\mathcal{S}$ of s rows from the input that have x as their antecedent. If the sampled rows are drawn uniformly, then finding the confidence of this sample will be a good estimate for C_x. More precisely, let $\hat{C}_x = \max_y |\{(x, y) \in \mathcal{S}\}|/s$. Many methods are known to draw a uniform sample $\mathcal{S}$ from a stream. In particular, the *reservoir sampling* method [27] will achieve this in a single pass. Applying standard sampling arguments yields the following result:

Algorithm 1 Two-pass

```
Require: relation $R$, CFD $(X \rightarrow Y, T_p)$
    $t$: number of rows to sample in pass 1
    $z$: reservoir size in pass 2
    variable: true for variable CFD, false for fixed CFD
$\mathcal{S}_R \leftarrow \emptyset$; {PASS 1}
for all rows $r_i$ in $R$ do
    if variable or $r_i[X] \asymp T_p[X]$ then
        Reservoir sample $r_i$ into $\mathcal{S}_R$ of size $t$;
for all $x$ such that $\exists r_j = (x_j, y_j) \in \mathcal{S}_R, x_j = x$ do
    $\mathcal{S}_x \leftarrow \emptyset$; {PASS 2}
for all rows $r_i = (x_i, y_i)$ in $R$ do
    if variable or $r_i[X] \asymp T_p[X]$ then
        if $\exists \mathcal{S}_x, x_i = x$ then
            Reservoir sample $r_i$ into $\mathcal{S}_x$ of size $z$
$\hat{C} = 0$; {Estimate Confidence}
for all stored $\mathcal{S}_x$ do
    if $x \not\asymp T_p[X]$ then
        Delete $\mathcal{S}_x$ from $\mathcal{S}_R$;
    else
        $\hat{C} \leftarrow \hat{C} + \frac{\max_{y|y \asymp T_p[Y]} |\{(x,y) \in \mathcal{S}_x\}| \cdot |\{r_i | r_i = (x_i, y_i) \in \mathcal{S}_R, x_i = x\}|}{|\mathcal{S}_x|}$;
return $\hat{C}/|\mathcal{S}_R|$;
```

LEMMA 1. *Given* x, *drawing a reservoir sample of* $s = O(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$ *rows uniformly from rows of the input that have* x *as their antecedent allows us to find* $\hat{C}_x$ *so that*

$$\Pr[|\hat{C}_x - C_x| > \epsilon] < \delta$$

3.1.2 Heavy Hitters Approach

The definition of confidence is based on finding which consequent occurs most commonly in the group of x. It therefore makes sense to look to *heavy hitters* algorithms to find such consequents. In particular, given x, we can feed all rows of R which have x as their antecedent into a streaming heavy hitters algorithm. This will return an approximation of the most frequent item in the stream, and an estimate of its frequency, $\hat{f}$. From this, we can set $\hat{C}_x = \hat{f}/N_x$. Picking an algorithm such as the deterministic *SpaceSaving* technique of Metwally *et al.* [25] yields the following result:

LEMMA 2. *Given* x, *running the SpaceSaving algorithm with space* $s = O(\frac{1}{\epsilon})$ *allows us to find* $\hat{C}_x$ *so that*

$$|C_x(R) - \hat{C}_x(R)| < \epsilon$$

3.1.3 Comparing Sampling and Heavy Hitters

The heavy hitters approach has tighter worst-case asymptotic bounds and provides a deterministic guarantee. However, empirically the sampling approach typically yields equally good results in practice using a similar amount of space. There are also cases where retaining a sample offers more flexibility than a heavy hitters data structure. In our experimental study, we evaluate sampling methods, and see them to be effective in practice.

3.2 Fixed CFD Estimation

Two-pass Sampling Algorithm. Our algorithm to estimate the confidence of a fixed CFD ϕ (recall Definition 4) takes two streaming passes over the input. In the first pass, we draw a sample $\mathcal{T}$ of t rows from the support set of ϕ uniformly, using reservoir sampling. For each group for x identified in the sample S, we make an estimate of the confidence C_x of the group as $\hat{C}_x$, using either approach from Section 3.1, with appropriate parameters (ϵ_g, δ_g). We take the t estimates and find their mean, and report this as the estimated confidence. That is, our final estimate $\hat{C}^\phi(R) = \sum_{x \in \mathcal{T}} \hat{C}_x/t$. The pseudocode for this algorithm is given in Algorithm 1 (note, in this implementation, if the same group is sampled multiple times in the first pass, only a single sample is made in the second pass, but its confidence is weighted up correspondingly in the final estimate).

THEOREM 1. *Sampling $t = 2\epsilon_c^{-2} \log \frac{2}{\delta}$ rows in the first pass yields an estimator which is within additive error $\epsilon = \epsilon_c + \epsilon_g$ with probability at least $1-\delta$. The total space required by this algorithm is $O(\epsilon^{-3} \log \frac{1}{\delta})$.*

PROOF. For the proof, we assume that the heavy hitters approach of Lemma 2 is used to find each estimate $\hat{C}_x$ with error at most ϵ_g. The output of the estimation process is a random variable, $\hat{C}$. The expectation of this random variable is

$$\begin{aligned} \mathsf{E}[\hat{C}] =& \frac{1}{t} \sum_x \mathsf{Pr}[x \in \mathcal{T}] \hat{C}_x = \sum_x \frac{N_x(C_x \pm \epsilon_g)}{N} \\ =& \sum_x \frac{\max_y N_{x,y} \pm \epsilon_g N_x}{N} = C \pm \epsilon_g \end{aligned}$$

Note that the estimator is formed as the mean of t repetitions, each of which has expected value C. By observing that each value of $\hat{C}_x$ is constrained to lie in the range $0 \ldots 1$, we can apply the Chernoff-Hoeffding bound to the mean of t repetitions:

$$\begin{aligned} & \mathsf{Pr}[\hat{C} - \mathsf{E}[\hat{C}] > \epsilon_c] < \exp(-2\epsilon_c^2/t) = \tfrac{1}{2}\delta \\ \text{and} \quad & \mathsf{Pr}[\mathsf{E}[\hat{C}] - \hat{C} > \epsilon_c] < \exp(-2\epsilon_c^2/t) = \tfrac{1}{2}\delta \end{aligned}$$

Combining these, we have that

$$\mathsf{Pr}[|\hat{C} - C| > (\epsilon_c + \epsilon_g)] < \delta,$$

so setting $\epsilon_c = \epsilon_g = \epsilon/2$ gives the result as required. □

Applying this algorithm to the "hard case" described in Section 2.3 ensures that the large group is picked up in the sample, in addition to many of the small groups, so the contribution of the large group is not lost. In the second pass, enough information is collected about each of the sampled small groups to correctly compute their contribution to the overall confidence.

3.3 Variable CFD Confidence Estimation

The more challenging problem of estimating the confidence of multiple CFDs from a single summary is stated formally in Definition 5. Although the attribute sets X and Y are fixed in advance, the tableau T_p is given after the data is seen, and can select an arbitrary set of groups. We adapt the above algorithm to this problem. If many rows satisfy T_p (i.e., the tableau has high support), then the above technique will sample many groups that satisfy T_p. If we base our estimate on the confidence of these groups and ignore samples from other groups, the uniformity of the sampling means that this is a uniform sample of such groups. Consequently, the resulting estimate of $C^\phi(R)$ should be accurate, providing the support of T_p is not too low.

THEOREM 2. *Running the two pass sampling algorithm using total space $O(\epsilon^{-3} \log \delta^{-1})$ enables us to estimate $C^\phi(R)$ with error $\epsilon/\sqrt{\text{supp}^\phi(R)}$.*

PROOF. We first assume that ϕ is assertion-free (that is, it does not place any requirements on the consequents), and later discuss how to remove this assumption. Let $\mathcal{T}$ be the sample of $t = O(\epsilon^{-2} \log 1/\delta)$ rows selected in the first pass of the algorithm, as before. Given a tableau T_p, let t' be the number of rows in the sample which satisfy $T_p[X]$. We have

$$\begin{aligned} \mathsf{E}[t'] &= \sum_{r_i \in R} \mathsf{Pr}[r_i \in \mathcal{T} \wedge r_i[X] \asymp T_p[X]] \\ &= \sum_{r_i \in R} \frac{t}{N} \text{supp}^\phi(R) = t \, \text{supp}^\phi(R) \end{aligned}$$

We argue that with high probability, t' will be close to its expected value. If the support of T_p is very small, then we obtain the required error guarantee by choosing our estimate to be $\hat{C}^\phi(R) = 0$. In particular, if $\text{supp}^\phi(R) \le \epsilon$ (and $\epsilon < 1$), then $\epsilon/\text{supp}^\phi > 1$, and so any estimate of the confidence is within the error bounds. Otherwise, $\text{supp}^\phi(R) > \epsilon$ and using the Chernoff-Hoeffding bound,

$$\begin{aligned} \mathsf{Pr}[t' < \text{supp}^\phi(R)t/2] &= \mathsf{Pr}[\mathsf{E}[t'] - t' > \text{supp}^\phi(R)t/2] \\ &\le \mathsf{Pr}[\mathsf{E}[t'] - t' > \epsilon t/2] \\ &\le \exp(-2(\epsilon^2 t^2/4)/t) \\ &\le \exp(-\epsilon^2 t/2) < \delta/4 \end{aligned}$$

which follows by our choice of t. So, apart from this small probability, we have at least $t \, \text{supp}^\phi(R)/2$ samples in the first pass which satisfy T_p. Applying the analysis of Theorem 1, this allows $C^\phi(R)$ to be estimated with additive error $\sqrt{4 \log(2/\delta)/t \, \text{supp}^\phi(R)} + \epsilon_g$. Ensuring $t \ge 4\epsilon_c^{-2} \log 2/\delta$ bounds the overall error by $\epsilon_c/\sqrt{\text{supp}^\phi} + \epsilon_g$ with probability at least $1 - \delta$. Observe that this guarantee, in terms of $\sqrt{\text{supp}^\phi}$, is actually stronger than that required by Definition 5.

An immediate consequence of this analysis is that we can estimate the support of the CFD, $\text{supp}^\phi(R)$, by t'/t. The Chernoff-Hoeffding bound argument above then shows that $\mathsf{Pr}[|t'/t - \text{supp}^\phi| > \epsilon] < \delta/2$, meaning that the sample also provides a good estimate of the support with additive error.

The above analysis requires only small modifications when the tableau is allowed to contain assertions. We first consider the case when reservoir sampling is being used. Given a sample of tuples which match on the antecedent, we can find a candidate keeper as the most frequent consequent in the sample which satisfies all the assertions in the tableau (if there are any such in the sample). For the same reasons as Lemma 1, the relative frequency of this keeper is a good estimate for the group confidence. Similarly, in the heavy hitters approach, it suffices to use the maximum frequency of any heavy hitter which satisfies all relevant assertions (if any). For the same reasons as the assertion-free case, the estimated frequency of this consequent estimates the group confidence with error at most ϵ_g. In both cases, if no candidate keepers are found, then the confidence of the group is estimated by 0. Putting this all together into the above proof, the same result follows. □

Lastly, we comment that our solution based on reservoir sampling is slightly more general. Note that the first pass depends only on the antecedent set X. If in the second pass we maintain a sample of *rows* for each sampled group, then this summary allows the estimation of the confidence of any CFD with embedded FD $X \rightarrow Y'$: the consequent Y' can be specified even at query time.

Algorithm 2 Idealized One-pass

```
Require: relation $R$, CFD $(X \to Y, T_p)$
    $t$: number of rows to sample
    $z$: reservoir size
    variable: true for variable CFD, false for fixed CFD
$\mathcal{S}_R \leftarrow \emptyset$; $n \leftarrow 0$;
for all rows $r_i = (x_i, y_i)$ in $R$ do
    if variable or $r_i[X] \asymp T_p[X]$ then
        for all $\mathcal{S}_j \in \mathcal{S}_R$ such that $x_i = x_j$ do
            Reservoir sample $r_i$ into $\mathcal{S}_j$ of size $z$
        $\mathcal{S}_i \leftarrow \{r_i\}$;
        Reservoir sample $\mathcal{S}_i$ into $\mathcal{S}_R$ of size $t$
$k \leftarrow 0$ ; $s \leftarrow 0$; {Estimate Confidence}
for all $\mathcal{S}_i \in \mathcal{S}_R$ do
    if $\mathcal{S}_i[X] \not\asymp T_p[X]$ then
        Delete $\mathcal{S}_i$ from $\mathcal{S}_R$
    else
        $s \leftarrow s + 1$;
        if $y_i \asymp T_p[Y]$ and $\forall y' \neq y_i |\{(x_i, y_i) \in \mathcal{S}_i\}| > |\{(x_i, y') \in \mathcal{S}_i\}|$ then
            $k \leftarrow k + 1$;
return $(k/s)$;
```

4. IDEALIZED ONE-PASS ALGORITHM

Drawing on ideas from the above algorithm, we now propose an "idealized" approach to estimating the confidence with only a single streaming pass over the relation R. We say "idealized", since it requires computing a quantity that can be hard in general to estimate accurately in one pass. The algorithm is based on multiple independent instances of a basic estimator. We first consider the fixed CFD estimation problem for assertion-free CFDs. Each estimator samples a row $r = (x, y)$ uniformly from the stream—using reservoir sampling, the algorithm does not have to know the length of the stream in advance to do this. We say that y is a keeper for this row if, in the suffix of the stream including the tuple, y is the strict mode for the group of x. More formally, if tuple $r_i = (x, y)$ is the ith tuple in the stream, then let R_i denote the suffix of the stream beginning at r_i. Let N_x^i denote the number of occurrences of x in the remainder of the stream, and $N_{x,y}^i$ be the number of occurrences of the tuple (x, y). Then y is a keeper for R_i if

$$\forall y' \neq y . N_{x,y}^i > N_{x,y'}^i.$$

Define the variable $X_i = 1$ if, for the ith row of the relation, $r_i = (x_i, y_i)$, y_i is a keeper for x_i in R_i, and 0 otherwise. We argue that if i is chosen uniformly from $[1 \ldots N]$, then this random variable X_i is an unbiased estimator for the confidence $C^\phi(R)$. First, we show that X_i can be written in terms of the number of keepers of different suffixes of the stream, then use this to prove the main claim.

LEMMA 3. $X_i = \max_y N_{x,y}^i - \max_y N_{x,y}^{i+1}$

PROOF. Consider $\max_y N_{x,y}^i - \max_y N_{x,y}^{i+1}$: if this is not zero, then it must be 1, since $\max_y N_{x,y}^i$ only considers one more row than $\max_y N_{x,y}^{i+1}$. The only way this quantity can be 1 is if the ith row from the relation contributes to an increase in the value of $N_{x,y}^i$ for some y. But this row $r_i = (x_i, y_i)$ can only increase the corresponding N_{x_i,y_i}. Therefore, the ith row (x_i, y_i) must have $N_{x_i,y_i}^{i+1} = \max_y N_{x_i,y}^{i+1}$ and hence $N_{x_i,y_i}^i > \max_y N_{x_i,y}^i$ for all $y \neq y_i$. But this corresponds precisely to the definition that y_i is a keeper for x_i in R_i. □

LEMMA 4. *The random variable X_i is an unbiased estimator for the confidence of the relation R. That is, $\mathsf{E}[X_i] = C^\phi(R)$.*

PROOF.

$$\mathsf{E}[X_i] = \sum_{i=1}^{N} \frac{X_i}{N} = \frac{1}{N} \sum_{i=1}^{N} \sum_{x} (\max_y N_{x,y}^i - \max_y N_{x,y}^{i+1})$$
$$= \frac{1}{N} \sum_{x} \max_y N_{x,y} = C^\phi(R)$$

□

Therefore, if we could find X_i exactly for each i in a sample, we could again sample $t = O(\epsilon^{-2} \log \frac{1}{\delta})$ tuples, and run this estimator to build an approximation with additive error at most ϵ. However, in general, finding this X_i value is hard, if there are many "near-keepers" for many suffices of the stream. That is, if for the group of x, there are several different consequents which have similar numbers of occurrences in every suffix of the stream, then it is (provably) hard to determine whether any given consequent is a keeper, which is required by the definition of X_i. Note though that in such cases the confidence of the group must be low, so we may be less interested in obtaining accurate estimates of the overall confidence in such cases. In practice, we may again use either a heavy hitter or sampling based method for each sampled row $r_i = (x_i, y_i)$ to determine whether y_i is a keeper for x_i in R_i. The pseudocode for this algorithm is given in Algorithm 2. From a formal perspective, this method is a heuristic, so we will evaluate it empirically and compare to the other methods we present.

When applying this algorithm to the "hard cases" in Section 2.3, there is an even chance that the sampled row r comes from the large group or one of the small groups. For the large group, there is a high chance that enough subsequent tuples indicate whether the confidence of the group is high or low. For a small group, either the first or second tuple in the group is sampled; if it is the first, then the second will be seen, and the confidence of the group correctly computed; in expectation, the result will be correct.

4.1 Extension to Variable CFD problem.

We first extend this approach to the variable tableau CFD confidence estimation problem in the case that T_p is assertion free. Again, we sample $t = O(\epsilon^{-2} \log 1/\delta)$ rows uniformly from the relation, and use the above procedure to estimate for each sampled row (x_i, y_i) whether y_i is a keeper for x_i. Again, define $X_i = 1$ if this is the case, and 0 otherwise. However, for our final estimate, we only compute the average of those sampled is for which $x_i \asymp T_p[X]$, and ignore the others. By the uniformity of the sampling, the sampled is correspond to a uniform sample of R', which consists of only those rows of R which satisfy T_p. Thus, we obtain an unbiased estimate of $C^\phi(R)$. Similarly to Theorem 2, we obtain a good estimate if sufficiently many estimates satisfy T_p. With probability at least $1 - \delta$, there are at least $\text{supp}^\phi(R)t/2$ such estimates, meaning that the obtained (idealized) estimate would have error $\epsilon/\sqrt{\text{supp}^\phi(R)}$. Since the first step of this algorithm is to draw a sample of tuples from the relation, it follows that we can also use this sample to estimate supp^ϕ, with the same argument as that in Theorem 2.

When T_p is not assertion free, we modify the estimator X_i so that it is 1 if and only if $y_i \asymp t_p[Y]$ for all t_p such that $x_i \asymp t_p[X]$, and $N_{x_i,y_i}^i > N_{x_i,y}^i$ for all $y \neq y_i$ that also satisfy the same condition. This still yields an estimator which is correct in expectation, but may be less accurate to determine from the summary, since restrictions on consequents mean there are fewer matching tuples.

Algorithm 3 Multi-level One-pass

```
Require: relation $R$, CFD $(X \to Y, T_p)$
    $N$: number of rows in relation $R$
    $\epsilon, \delta$: accuracy parameters
    variable: true for variable CFD, false for fixed CFD
$b \leftarrow \frac{1}{\epsilon^3}\log\frac{1}{\delta}$; $l_{\max} \leftarrow \log N$;
for $k = 0$ to $l_{\max}$ do
    $\alpha_k \leftarrow \min(\frac{1}{2^k \epsilon^2}, 1)$; $n_k = 0$;
    Initialize $hash_k$ with range $[0, 1)$;
    Initialize $CMGroup_k$ sketch with parameter $b$;
    Initialize $CMTuple_k$ sketch with parameter $b$;
for all rows $r_i = (x_i, y_i)$ in $R$ do
    if variable or $r_i[X] \asymp T_p[X]$ then
        $n = n + 1$;
        for $k = 0$ to $l_{\max}$ do
            if $hash_k(x_i) \le \alpha_k$ then
                $n_k = n_k + 1$;
                Update $CMGroup_k$ with $x_i$;
                Update $CMTuple_k$ with $(x_i, y_i)$;
$r$ = a random number $\in [1, 2]$; {Estimate Confidence}
for $k = 0$ to $l_{\max}$ do
    $keepers_k \leftarrow 0$;
    $HeavyGrp_k \leftarrow \epsilon$-heavy hitters from $CMGroup_k$;
    $HeavyTpl_k \leftarrow \epsilon$-heavy hitters from $CMTuple_k$;
    for all $(x, n_x) \in HeavyGrp_k$ such that $x \asymp T_p[X]$ do
        if $\frac{r}{2^{k+1}} < \frac{n_x}{n} \le \frac{r}{2^k}$ then
            $keepers_k \leftarrow keepers_k +$
            $\max_y\{n_{x,y} | ((x, y), n_{x,y}) \in HeavyTpl_k, y \asymp T_p[Y]\}$;
return $\frac{1}{supp(R)\cdot n}\sum_0^{l_{\max}} \frac{keepers_k}{\alpha_k}$;
```

5. ONE PASS ALGORITHM BASED ON MULTI-LEVEL SAMPLING

In this section, we describe a one-pass algorithm for the fixed CFD estimation problem. The algorithm and its analysis are more involved than the previous methods, in order to give the desired guarantees in a single pass.

5.1 One Pass Algorithm

The algorithm builds summaries for different "levels" k, for each k in the range $0 \dots \log N$: At level k, each group is initially sampled with probability $\alpha_k = \min\{2^{-k}\epsilon^{-2}, 1\}$ — note that for $k \le 2\log_2 1/\epsilon$, all groups are sampled, and so can be treated together. The group sampling is accomplished using min-wise hashing [5]: a hash function is applied to the antecedent x mapping it into the range $0 \dots 1$, and is sampled if the hash value is at most α_k. If a group is sampled at level k, then it is inserted into a Count-Min sketch with $b = O(\epsilon^{-3}\log 1/\delta)$ buckets based on the group ids. In parallel, we also insert the (x, y) pairs that are sampled at level k into a Count-Min sketch with $b = O(\epsilon^{-3}\log 1/\delta)$ buckets based on the (x, y) values. We use the sketch of groups at each level to extract the 'heavy hitter' groups (those that account for more than an ϵ fraction of the frequency at that level), along with their corresponding estimated $\hat{N}_x$ values. From the tuple sketch at each level, we extract the heavy hitter consequents, and their corresponding estimate $\hat{N}_{x,y}$ values.

The final estimation process picks a randomly chosen "shift" $r \in [1, 2]$ to avoid some adversarial cases. For a heavy group x recovered at level k, we test whether its estimated support falls in the range $\frac{r}{2^k+1} < \frac{\hat{N}_x}{N} \le \frac{r}{2^k}$. If so, we estimate its $\max_y \hat{N}_{x,y}$ value using the sketch of tuples. From these, we compute

$$\hat{C}^{\phi}(R) = \sum_{k=0}^{\log N} \frac{1}{\alpha_k N} \sum_{x \text{ at level } k} \max_y \hat{N}_{x,y}.$$

We claim that the resulting estimator solves the fixed CFD estimation procedure by producing an estimate of the confidence $C^{\phi}(R)$ with additive error at most ϵ.

For example, on the "hard case" given in Section 2.3, the large group would be dominate for low values of k, and its statistics recovered at these levels. But for higher values of k, the large group is unlikely to be sampled, allowing the (many) small groups more chances to be sampled, and statistics collected. Combining the information from all levels of the sampling could then provide the overall confidence. Pseudocode for the algorithm is given in Algorithm 3.

5.2 Analysis

Define G_k as the set of groups with support close to $r2^{-k}$ as:

$$G_k = \{x : \frac{r}{2^{k+1}} < \frac{N_x}{N} \le \frac{r}{2^k}\}$$

First, we show that if we could draw a uniform random sample of the groups and find the corresponding $N_{x,y}$ values exactly, we would have an estimator that is correct in expectation and with bounded variance. Let S_k be the groups sampled at the kth level (so they are picked with probability α_k). Let C_x be the confidence of the group x as before, so that $C_x = \frac{\max_y N_{x,y}}{N}$ and $C^{\phi}(R) = \sum_x C_x$. Define the estimator X_k as

$$X_k = \sum_{x \in (S_k \cap G_k)} \frac{C_x}{\alpha_k}$$

LEMMA 5. *The expectation and variance of X_k satisfies*

$$\mathsf{E}[X_k] = \sum_{x \in G_k} C_x \quad \textit{and} \quad \mathsf{Var}[X_k] = \epsilon^2 \mathsf{E}[X_k]\ .$$

PROOF. For the expectation,

$$\mathsf{E}[X_k] = \mathsf{E}\left[\sum_{x \in (S_k \cap G_k)} \frac{C_x}{\alpha_k}\right] = \sum_{x \in G_k} \mathsf{Pr}[x \in S_k]\frac{C_x}{\alpha_k} = \sum_{x \in G_k} C_x.$$

For the variance, define $I_{x,k}$ as the indicator variable that is 1 if x is chosen to be in the sample S_k, and 0 otherwise. Then

$$\begin{aligned}\mathsf{Var}[X_k] &= \sum_{x \in G_k} \mathsf{Var}\left[I_{x,k}\frac{C_x}{\alpha_k}\right] = \sum_{x \in G_k} \frac{C_x^2}{\alpha_k^2}\mathsf{Var}[I_{x,k}] \\ &= \sum_{x \in G_k} \frac{C_x^2}{\alpha_k^2}(\alpha_k)(1 - \alpha_k) \\ &\le \sum_{x \in G_k} \frac{C_x 2^{-k}}{\alpha_k} = \epsilon^2 \mathsf{E}[X_k]\ .\end{aligned}$$

□

This shows that if we could estimate each C_x value of the sampled groups accurately, then these could be combined into an estimate with low variance for the overall confidence. In particular, for $X = \sum_{k=0}^{\log N} X_k$, we have $\mathsf{Pr}[|X - C^{\phi}(R)| > \epsilon] \le \delta$, by the Chebyshev inequality. We now show that by using the Count-Min sketch summary, each C_x is approximated with high accuracy.

LEMMA 6. *Using Count-Min sketches with $b = 2\epsilon^{-3}\log 1/\delta$ buckets allows N_x and $N_{x,y}$ to be estimated with additive error $\epsilon^3\alpha_k N$.*

PROOF. Given any group x in S_k, the two Count-Min sketches with b buckets promise to estimate N_x and $N_{x,y}$ with error proportional to $\Delta = F_1^{res(b)}/b$, where $F_1^{res(b)}$ is the frequency of all items in S_k excluding the b most frequent items [13]. Note that if N_x or $N_{x,y}$ is less than Δ, then estimating it by 0 will satisfy this requirement.

We now analyze $F_1^{res(b)}(S_k)$, and argue that it is at most $10\alpha_k N \log 1/\delta$ with probability at least $1-\delta$. First, consider groups x such that $N_x \geq 2^{-k}N$ — there can be at most 2^k of these. Moreover, in expectation, the random sampling picks at most $\alpha_k 2^k = \epsilon^{-2}$ such groups. Using the relative Chernoff bound, which states that for $\beta > 2e-1$,

$$\mathsf{Pr}[X > (1+\beta)\mathsf{E}[X]] < 2^{-\mathsf{E}[X]\beta},$$

we have that the probability of picking more than $b = \epsilon^{-3}\log\frac{1}{\delta}$ groups is at most $2^{-\epsilon^{-2}(\epsilon^{-1}\log\delta^{-1}-1)} < \delta$ for $\delta, \epsilon < 1/2$.

For groups with $N_x < 2^{-k}N$, we bound the total *frequency* that these contribute. Since these are sampled with probability α_k, the expected sum of frequency of all such sampled items is at most $\alpha_k N$. Applying the relative Chernoff bound again, and using the fact that $\alpha_k \leq 1$, the probability that $F_1^{res(b)}(S_k) > 10\alpha_k N \log 1/\delta$ is at most δ. The analysis of $F_1^{res(b)}(S_k)$ for antecedent, consequent pairs $N_{x,y}$ is similar. The difference arises since the sampling choice is based only on the antecedent x, so there are correlations between consequents in the same group. However, since the analysis is based on expectations, these do not affect the result, and again, we can conclude that the $F_1^{res(b)}$ is bounded. □

Using the sketches at level k, we recover $\hat{N}_{x,y}$ values for all x for which $x \in \tilde{G}_k$, where $\tilde{G}$ is defined as

$$\tilde{G}_k = \{x : r2^{-k-1} < \frac{\hat{N}_x}{N} \leq r2^{-k}\}.$$

The above lemma shows that these estimates are quite accurate. If we replace the random variable X_k with $\tilde{X}_k$, the same estimator built with the estimated frequencies, we obtain:

$$|X_k - \tilde{X}_k| \leq \frac{1}{\alpha_k N} \sum_{x\in S_k\cap(G_k\cap\tilde{G}_k)} |\max_y N_{x,y} - \max_y \hat{N}_{x,y}| + \sum_{x\in S_k\cap(G_k\setminus\tilde{G}_k)} \max_y N_{x,y} + \sum_{x\in S_k\cap(\tilde{G}_k\setminus G_k)} \max_y \hat{N}_{x,y}$$

That is, the error comes from three places: error in the estimated frequencies on antecedents that both methods agree should be in the sample, plus groups missed by the sketching approach that should be included, and groups included by the sketching approach that should be omitted. For the first of these, we have

$$\begin{aligned}\frac{1}{\alpha_k N} \sum_{x\in S_k\cap(G_k\cap\hat{G}_k)} &|\max_y N_{x,y} - \max_y \hat{N}_{x,y}| \\ &\leq \sum_{x\in S_k\cap G_k} \frac{1}{\alpha_k N}\epsilon^3\alpha_k N \\ &\leq \sum_{x\in S_k\cap G_k} \epsilon^3 \leq \epsilon \sum_{x\in G_k} \frac{N_x}{N}\end{aligned}$$

The last equality holds in expectation because for $x \in G_k$, we have $\alpha_k \leq 2^{-k}\epsilon^{-2} \leq \frac{N_x}{N\epsilon^2}$. Since each x is sampled into S_k with probability α_k, the expected value of this summation is $\epsilon\sum_{x\in G_k} N_x/N$. Observe that if we take this sum over all x, then $\epsilon\sum_x N_x/N \leq 2\epsilon$ with constant probability, applying the Markov inequality.

For the errors due to misclassification of antecedents into groups, these affect only those x for which

$$\begin{aligned}N_x &\in [r2^{-k}N - r\epsilon^3\alpha_{k+1}N, r2^{-k}N] = [(1-2\epsilon)r2^{-k}N, r2^{-k}N] \\ \text{or } N_x &\in [r2^{-k}N, r2^{-k} + r\epsilon^3\alpha_k N] = [r2^{-k}N, (1+\epsilon)r2^{-k}N]\end{aligned}$$

For other values of N_x, the estimated $\hat{N}_x$ will place it in the correct G_k by the guarantees on the sketch estimation. We can bound this quantity by observing that, based on the random choice of the parameter r,

$$\mathsf{E}[\sum_{x\in[(1-2\epsilon)r2^{-k}N,(1+\epsilon)r2^{-k}N]} N_x] = 3\epsilon N/2^k.$$

So the contribution to the confidence of all such groups is at most 3ϵ with constant probability, and errors on these groups can contribute at most this much to the overall error in the confidence estimate.

In conclusion then, our final estimate $\hat{C}^\phi(R)$ formed as the sum of all estimates $\tilde{X}_k$ satisfies:

$$|\hat{C}^\phi(R) - X| \leq \sum_{k=0}^{\log N} |\tilde{X}_k - X_k| \leq (\sum_{k=0}^{\log N} 2\epsilon \sum_{x\in G_k} \frac{N_x}{N}) + 3\epsilon = 5\epsilon$$

with constant probability. We amplify this to arbitrarily small probability of failure by repeating the estimation process (with different random choices of the shift r) sufficiently many times, and taking the median of the estimates. Thus, by the bounds on X, we conclude (after appropriate rescaling of ϵ) that:

THEOREM 3. *The estimate $\hat{C}^\phi(R)$ produced by the multilevel algorithm satisfies*

$$\mathsf{Pr}[|C^\phi(R) - \hat{C}^\phi(R)| > \epsilon] < \delta\ .$$

5.3 One-pass Variable CFD estimation

We now outline how to use the above data structure to estimate the confidence of a CFD given after the data has been seen. Given the same set of sketches, we again extract the heavy hitter groups and their estimated N_x values at each level. But now we consider only those x values for which $x \asymp T_p[X]$. For these xs, we estimate the heavy hitter tuples to find the corresponding $N_{x,y}$ values, and pick the largest of these which matches the requirements on the consequent (if any). We also need to estimate the support of ϕ, which can be done by additionally keeping a uniform sample of rows. We then estimate

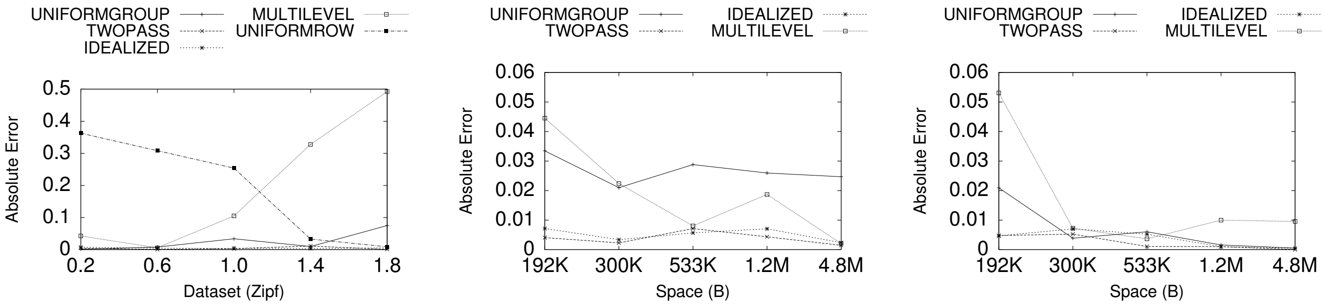
$$\hat{C}^\phi(R) = \sum_{k=0}^{\log N} \frac{1}{\alpha_k \widehat{\text{supp}}^\phi(R) N} \sum_{x\asymp T_p[X]\text{at level } k} \max_{y\asymp T_p[Y]} N_{x,y}$$

The analysis of correctness relies on the linearity of expectation. We omit full details in this presentation for brevity.

6. LOWER BOUNDS FOR CONFIDENCE ESTIMATION

We show that there are certain fundamental limitations on what is possible for any algorithm which takes a small number of passes over the input to estimate the confidence of a fixed CFD.

THEOREM 4. *Estimating $C^\phi(R)$ with relative error $\epsilon < 1/3$ in P passes over input of size N with at most constant probability of error requires at least $\Omega(N/P)$ space*



(a) Accuracy on synthetic data (b) Accuracy on Syn1 with $z = 0.8$ (c) Accuracy on Retailer data set

Figure 1: Accuracy of algorithms in estimating the confidence of a fixed CFD

PROOF. We show a reduction from the DISJOINTNESS problem in communication complexity [24]. In this problem, two players (Alice and Bob) each hold a binary vector (a and b, respectively) of length n. The problem is to determine whether there is any index i such that $a[i] = b[i] = 1$, or if there is no such index. It is known that determining the answer in P messages requires Alice and Bob to exchange $\Omega(n)$ bits of communication in total.

Given vectors a and b, we can create an instance of the fixed CFD estimation problem. The instance has a single group, with some fixed antecedent, x, say. For each entry in a such that $a[i] = 1$, we insert (x, i) at the start of the constructed relation. Similarly, for each i such that $b[i] = 1$, we insert (x, i) at the end of the constructed relation. This generates a relation R with N rows. Now consider the confidence $C^{\phi}(R)$, where the tableau contains a single wildcard-only pattern: if the two vectors are disjoint, then $C^{\phi}(R) = 1/N$. But if they have an intersection, then $C^{\phi}(R) = 2/N$. A relative error estimate would find $\hat{C}^{\phi}(R)$ so that $(1 - \epsilon)C^{\phi}(R) \le \hat{C}^{\phi}(R) \le (1 + \epsilon)C^{\phi}(R)$. So if we could estimate $C^{\phi}(R)$ with relative error better than $\epsilon = 1/3$, then we could distinguish these two cases. An efficient algorithm with P *passes* to estimate the confidence could be turned into an efficient communication protocol with $2P$ *messages*: Alice would run the algorithm of the part of the relation corresponding to her string, then send a message consisting of the current memory state of the algorithm to Bob, who would continue the execution on the portion of the relation derived from his string, and so on. From the communication necessary for the DISJOINTNESS problem, it follows that at least $\Omega(n/P) = \Omega(N/P)$ space must be needed for any algorithm estimating the confidence of a CFD with P passes. □

This shows that we should not expect to design algorithms which can give answers with relative error. But note that the hard case at the heart of the above proof corresponds to a very low confidence. In general, we are interested in cases when the confidence is at least some constant value (say, at least 0.1). In these cases, an *additive error* (as required in Definition 4) will suffice. Even here, there are limits to how accurately the problem can be solved.

THEOREM 5. *Solving the fixed CFD estimation problem with additive error at most ϵ in a constant number of passes over the data with at most constant probability of error requires at least $\Omega(\frac{1}{\epsilon^2})$ space.*

PROOF. We show a reduction from the GAPHAMMING problem in communication complexity. As in the previous theorem, Alice and Bob each hold a binary vector of length n (a and b respectively). In addition, there is the promise on the Hamming distance H between the vectors so that either $H(a, b) \le \frac{n}{2} - \sqrt{n}$ or $H(a, b) \ge \frac{n}{2} + \sqrt{n}$. It is known that determining which case holds with a constant number of messages between Alice and Bob requires at least $\Omega(n)$ bits of communication [21, 6].

Given such vectors, we can create an instance of the CFD estimation problem. For each a_i, we create an (antecedent, consequent) pair (i, a_i); and similarly, for each b_i we create the pair (i, b_i). This creates a relation R of $N = 2n$ pairs. Now observe that the confidence of this relation (again, assuming a tableau with a single all-wildcards pattern) is

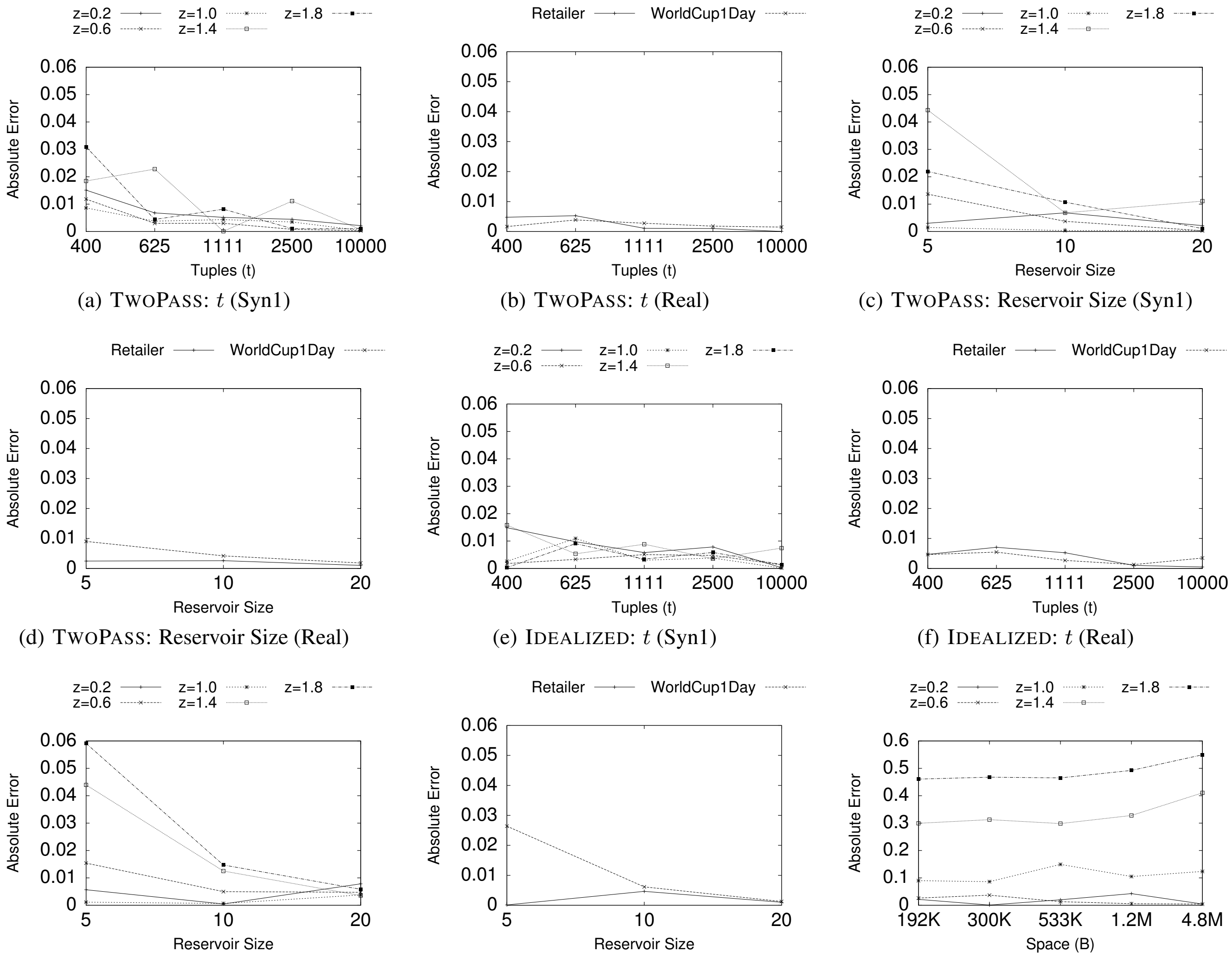
$$\sum_x \frac{N_x}{N} \max_y \frac{N_{x,y}}{N_x} = \frac{2}{N}\left(\frac{N}{2} - \frac{H(a,b)}{2}\right) = 1 - \frac{H(a,b)}{N}$$

By the promise on the Hamming distance between a and b, this quantity is either less than $\frac{3}{4} - \frac{1}{\sqrt{2N}}$ or more than $\frac{3}{4} + \frac{1}{\sqrt{2N}}$. So if we set $\epsilon = \frac{1}{\sqrt{2N}}$ we could determine which case holds. As in the previous theorem, any algorithm taking a constant number of passes over the relation and storing a small amount of data can be transformed into an efficient communication protocol: Alice runs the algorithm on the portion of the relation derived from her bitstring, then send the memory contents to Bob who would continue on the tuples from his bitstring, and so on. Since this would solve the GAPHAMMING communication problem, it follows that a data structure of size $\Omega(N) = \Omega(\frac{1}{\epsilon^2})$ bits is needed to solve this problem, even allowing a constant probability of error. □

7. EXPERIMENTAL STUDY

We conducted an empirical study of the three proposed algorithms for CFD confidence estimation on both real and synthetic data sets. The first real data set *Retailer* contains 3×10^5 sales records from an online retailer (this is the largest data set used in [11], where it is described in more detail). The schema is {`type`, `name`, `country`, `price`, `tax`, `itemid`}. The second real data set *WorldCup1Day* and the third real data set *WorldCup1Month* are part of the 1998 World Cup website access logs [1]. *WorldCup1Day* contains the access log during a consecutive 13 hour time span, corresponding to the first 7×10^6 records from day 69. *WorldCup1Month* contains the access log during a consecutive 32 day time span, from the start of day 49 to the end of day 79, and has 10^9 records. The schema is {`objectID`, `day`, `hour`, `size`, ...}, where `objectID` is the ID of the object in each request and `size` is the number of bytes in the response.

Each synthetic data set contains 10^6 records with the same schema as the *Retailer* data set. The group sizes are chosen based on a Zipfian distribution, with parameter z. The confidence of each group is chosen by picking a value C_x in $[0, 1]$ uniformly at random. In the first class of synthetic data sets (*Syn1*), for each group,



(a) TWOPASS: t (Syn1)

(b) TWOPASS: t (Real)

(c) TWOPASS: Reservoir Size (Syn1)

(d) TWOPASS: Reservoir Size (Real)

(e) IDEALIZED: t (Syn1)

(f) IDEALIZED: t (Real)

(g) IDEALIZED: Reservoir Size (Syn1)

(h) IDEALIZED: Reservoir Size (Real)

(i) MULTILEVEL: Space (Syn1)

Figure 2: Sensitivity to Parameters

one consequent is chosen to achieve the desired confidence C_x, and all other consequents are chosen to be distinct. In the second class (*Syn2*), one consequent is chosen to have confidence C_x, and $\lfloor 1/C_x \rfloor$ others are chosen to have confidence as close to C_x as possible. This is chosen to be especially challenging for the idealized one-pass algorithm, since there are many "near keepers" in these groups. The default CFD used for *Syn1, Syn2*, and *Retailer* is the embedded FD {`type`, `name`, `country`} → {`price`, `tax`, `itemid`}, with the (fixed) tableau ('book'|__|__||__|__|__) (denoted *CFD1*). The true confidence of *CFD1* over *Retailer* is 0.908.

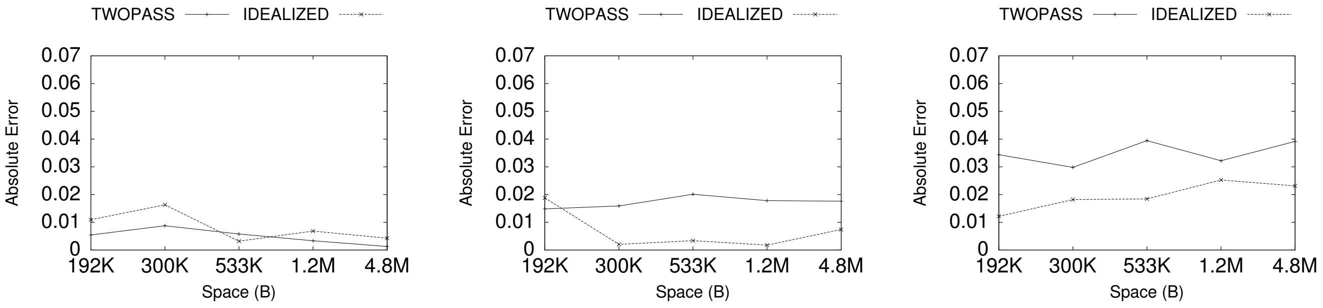
The embedded FD used for *WorldCup1Day* (*CFD2*) is {`objectID`, `day`, `hour`} → {`size`}, with the tableau (__|69|__||__), unless otherwise stated. The group sizes in both real data sets approximately follow a Zipfian distribution: for *WorldCup1Day*, the skewness parameter is close to $z = 1.8$, and for *Retailer*, $z = 0.4$. The true confidence of *CFD2* over *WorldCup1Day* is 0.88.

A third real dataset *WorldCup1Month* with one billion records is used to demonstrate the scalability in addition to other properties of the relevant algorithms. Its embedded FD (*CFD3*) is {`objectID`, `day`} → {`size`}, with the pattern (__|__||__). The true confidence of *CFD3* over *WorldCup1Month* is 0.78.

As two baseline algorithms, we also implement UNIFORMROW and UNIFORMGROUP, as discussed in Section 2.3. UNIFORMROW collects a uniform random sample of tuples and uses the confidence of the CFD in the relation induced by the sample as the final estimate. UNIFORMGROUP collects a uniform random sample of groups, and a uniform random sample of consequents for every sampled group. The estimate is the average confidence of all sampled groups.

7.1 Implementation Issues

We implemented all the algorithms in C++ and ran experiments on a shared machine with Intel Xeon 2.83GHz dual CPU running RedHat Linux with 16GB RAM. In the first pass, the TWOPASS algorithm used reservoir sampling to collect a uniform random sample of rows. The identifiers of the sampled groups are stored in hash tables for quick testing in the second pass. The IDEALIZED algorithm also used reservoir sampling to determine which groups to sample, and again to determine whether the sampled row constitutes a keeper for the suffix of the stream. The MULTILEVEL



(a) Syn2 $z = 0.4$ (b) Syn2 $z = 0.8$ (c) Syn2 $z = 1.8$

Figure 3: Accuracy comparison between TwoPass **and** Idealized

algorithm used Count-Min sketches based on the reference implementations available from `http://www.research.att.com/~marioh/frequent-items`. The default values of the number of samples, t, is 2500 in TwoPass and Idealized. When estimating the confidence of a single group, the default reservoir size is 20.

7.2 Fixed CFD Estimation

We begin by comparing the performance of all five algorithms mentioned above under the same space usage in estimating the confidence of fixed CFDs. Figure 1(a) plots the absolute error in the estimated confidence value on synthetic data sets (*Syn1*) with z ranging from 0.2 to 1.8 when each algorithm is allocated 1.2MB. On all data sets, TwoPass and Idealized outperform the other three and show little variation as the skew of the data is increased. They consistently obtain an error of less than 1% over all synthetic data sets. UniformGroup and MultiLevel have satisfactory performance when the data set is rather uniform. As the data set becomes more skewed, their errors grow. Since TwoPass and UniformGroup use the same parameters (same number of sampled groups, and same reservoir size to estimate the confidence of each group) the performance discrepancy between these two algorithms is a consequence of the representativeness of groups chosen and the different weighting schemes. Due to its high error, we do not consider UniformRow further: it gives poor results in all experiments tried. Meanwhile, MultiLevel does tend to show better results when given more space, as we see in later experiments.

Figures 1(b) and 1(c) illustrate the performance of each algorithm under different space usage on data with Zipfian group size distribution ($z = 0.8$) from (*Syn1*), and *Retailer* respectively. On both data sets, as they are given more space, the performance of all four algorithms tends to improve. TwoPass and Idealized are able to give good estimates even with very limited space (a few hundred kilobytes), and so show the least room for improvement with increasing space. MultiLevel is able to give good estimates when there is enough space for accurate Count-Min sketches. It achieves 1% error on the real data set *Retailer* when given space of the order of a megabyte (although *Retailer* is the smallest data set considered, it still requires 11MB to store in full). Notice that even though UniformGroup results in low error (< 0.01) in moderate space ($\geq$ 300K) in Figure 1(c), it never achieves absolute error below 0.02 on the synthetic data in Figure 1(b). This is because the group distribution in *Retailer* is rather uniform, which is the best case for UniformGroup. On the WorldCup data sets (not shown), it did appreciably worse: for the default setting of parameters, the absolute error is around 0.05.

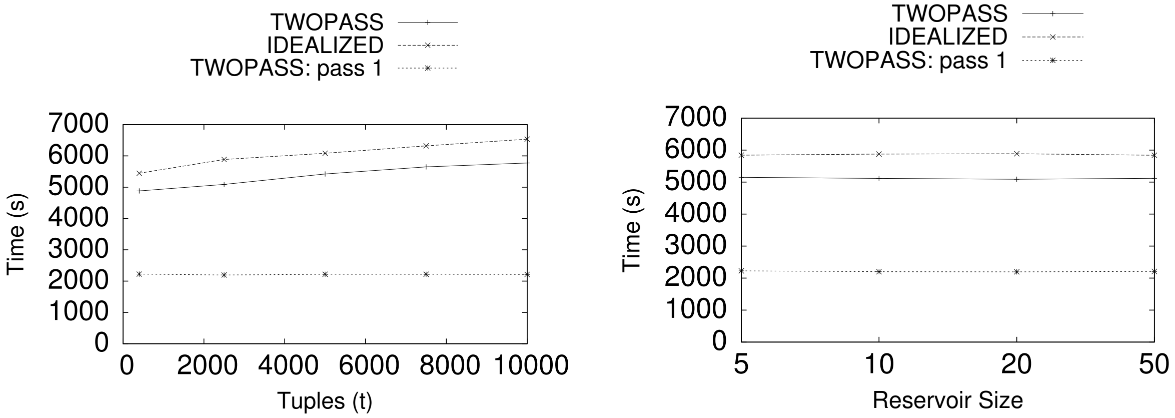
Sensitivity to Parameters. Our second set of experiments studies the influence of the various parameters for each of the proposed algorithms in turn. The results are illustrated in Figure 2. Figure 2(a) shows the impact of varying the main parameter of TwoPass, the number of tuples t sampled in the first pass, over synthetic data. This shows a general trend of increasing accuracy as t increases, as predicted by the analysis. A similar pattern occurs on the real data (Figure 2(b)), although with lower overall error (figures are shown on the same y-scale for ease of comparison). The size of the reservoir allocated to each group to estimate the confidence can be quite small and still obtain high accuracy: Figures 2(c) and 2(d) show that with only tens of samples per group, the overall accuracy is estimated well. Doubling the reservoir size seems to roughly halve the observed error, but beyond a reservoir size of 20 there is little room for improvement.

We see a similar trend for the number of tuples t sampled by the Idealized algorithm in Figure 2(e) and 2(f). The broad trend is for decreasing error as t increases, which is easier to see when viewing this algorithm in isolation. Likewise, doubling the size of the reservoir used to estimate whether the sampled tuple is a keeper seems to (more than) halve the observed error, though of course there are still some random fluctuations due to the random nature of the algorithms (Figures 2(g) and 2(h)).

Lastly, for MultiLevel, there is only a single parameter affecting the algorithm, which is the size of the Count-Min sketches used. Figure 2(i) plots the observed accuracy as a function of the total space used by the algorithm. This clearly shows the behavior of this algorithm is dominated by the skew of the data: for highly skewed data, it has large error. In fact, the plots also show that all three algorithms do seem to vary somewhat with the skew of the data: in all cases, the more skewed the data, the harder it seems to be to estimate the confidence. However, for TwoPass and Idealized, when these algorithms are given sufficient space, these differences become very small indeed.

7.3 Two-pass vs Idealized One-pass Algorithm

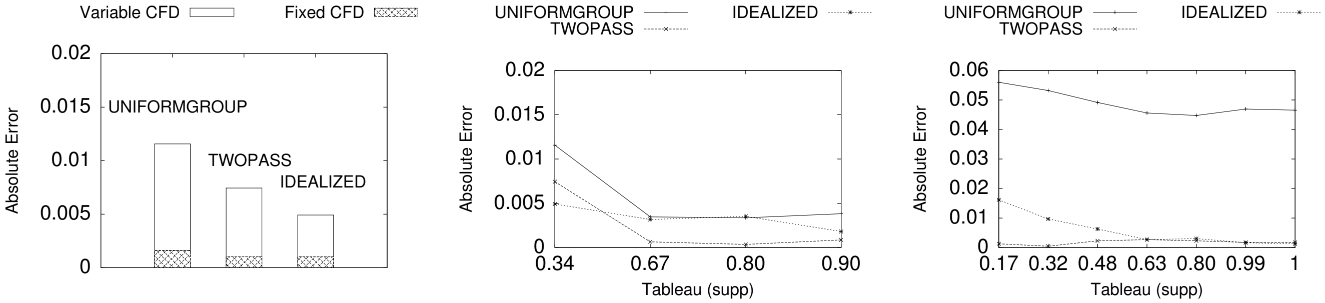
Since TwoPass and Idealized have the best performance over a variety of data sets and space parameters, we compare them more closely. Figure 3 shows more detail on three synthetic data sets from (*Syn2*)—a data set designed to be more challenging for Idealized. In fact, this data is a challenge for both algorithms, and as the skew increases, the errors increase, with TwoPass achieving worse results. In part, this may be due to the fact that the confidence is dominated by a few large groups: even if these are all sampled in the first pass, a reservoir of size 20 is used to estimate their confidence in our experiments. So only a small portion of



(a) Time: t (WorldCup1Month)

(b) Time: Reservoir Size (WorldCup1Month)

Figure 4: Performance comparison of TWOPASS **and** IDEALIZED



(a) Fixed and Variable CFD (Retailer)

(b) Confidence Estimation (Retailer)

(c) Confidence Estimation (WorldCup1Day)

Figure 5: Variable Tableau CFD Confidence Estimation

summary is being actively used to estimate the confidence in these cases, suggesting room for optimization.

Figure 4 illustrates the running time of the two algorithms on the *WorldCup1Month* data set of 10^9 records. The size of this dataset makes it impractical to compute the confidence exactly—we ran a series of group-by queries that compute the confidence on this data set. After 12 hours, the computation had not concluded, and was terminated. Analytically, the cost should be approximately linear in t, the number of tuples initially sampled by the two algorithms. As shown in Figure 4(a), this is close to the truth: the cost displays a slightly sublinear trend. This is partly because there is more opportunity for sharing information when the same group is sampled multiple times. Generally speaking, Figure 4(a) shows a slow increase in the running time with the increase of t. Given the size of *WorldCup1Month*, the cost of maintaining a random sample of t tuples using reservoir sampling becomes significant. Since the time for reservoir sampling does not directly depend on the size of the reservoir, this cost does not vary proportionally with t. To see this, we break the cost of TWOPASS down further by separating the cost of the first pass: it is clear that the cost of maintaining a random sample of t tuples remains stable regardless of t. The slightly increasing trend in both algorithms can be explained by the information collection for the sampled t suffixes/groups. This part of the cost grows proportionally with t.

We also observe that the time cost does not significantly vary with the size of the reservoir used, as shown in Figure 4(b). This is to be expected since, again, the time for the reservoir sampling does not directly depend on the size of the reservoir. In total, both algorithms take about 1.5 hours to process a data set of 10^9 rows. The time cost of extracting an estimated confidence from the summaries is also minimal: too small to measure in most cases.

7.4 Variable CFDs

Our final set of experiments are on the variable tableau CFD confidence estimation problem, where an arbitrary tableau can be given after the data has been seen. We compare the efficacy of TWOPASS, IDEALIZED and UNIFORMGROUP on instances of this problem. Figure 5(a) shows the results of these algorithms when given the same amount of space and a CFD ϕ on *Retailer* data with support value $\text{supp}^{\phi}(\textit{Retailer}) = 0.34$. All three algorithms give a larger error compared to the fixed CFD (i.e. when the support is 100%). In particular, the increase in the error of UNIFORMGROUP is much larger than that of the other two methods. Figure 5(b) shows the error of TWOPASS and IDEALIZED when estimating the confidence of variable CFDs with varying support. It displays a trend of increasing error as the support decreases, broadly in line with that predicted by the analysis of these algorithms. TWOPASS outperforms IDEALIZED in the majority of cases, which is consistent with our conclusion in Section 7.3. Similar results are seen in Figure 5(c) on the *WorldCup1Day* data set: here, the UNIFORMGROUP approach is more clearly seen to behave significantly worse than the proposed algorithms by large factors.

7.5 Experimental Summary

From our experimental analysis, we draw the following conclusions:

- TwoPass and Idealized are able to provide estimates with a very small error given very limited space, over the whole range of skew values observed in the data sets. Their relative improvement in accuracy over UniformRow can be up to 2 orders of magnitude, and up to 1 order of magnitude over UniformGroup when the data set is more skewed.

- MultiLevel, while having strong analytical guarantees, requires a large amount of space in order to produce good estimates in practice. This is due to needing to store many large Count-Min sketches for sufficient accuracy.

- Increasing the space available to the algorithms (such as increasing the number of samples, or the reservoir size) tends to improve the accuracy of all three proposed algorithms.

- TwoPass is slightly more accurate than Idealized in some cases. This comes at the price of requiring two passes over the data instead of one.

- Both TwoPass and Idealized do a good job of solving the variable tableau CFD estimation problem, and in particular are seen to produce results that are appreciably better than UniformGroup. Both are able to process data at a rate of over 10^5 tuples per second.

8. RELATED WORK

The problem of estimating the confidence of CFDs is related to work on finding conditional functional dependencies, estimating the confidence of functional dependencies and other integrity constraints, and generating concise summaries of large data sets. Recent research on CFDs has followed three directions. The first involves reasoning about CFDs, that is, axiomatization, consistency, and implication [3, 4, 14, 16]. The second employs exact and approximate CFDs to characterize the semantics of data, for example, discovering and verifying the confidence of CFDs on a given relation instance [10, 14, 15, 18]. The third uses CFDs for data cleaning, such as identifying and "repairing" tuples in a given relation instance that violate a given set of CFDs [11], which is usually performed after an appropriate set of CFDs has been identified and validated. Our work is closest to the second of these categories, although no prior work has addressed problems of estimating the confidence of CFDs directly. Our techniques for confidence estimation may be used in lieu of exact validation via an expensive SQL query, or when the entire data set cannot be stored or readily accessed.

The related question of determining whether an FD holds has been studied in [23]. In particular, it was shown that if a uniform random sample of size $O(\epsilon^{-1}\sqrt{N})$ is collected, then any FD that does not hold perfectly on the sample cannot (with high probability) have a confidence of above $1 - \epsilon$. This was used to determine which FDs could hold exactly, by eliminating any which did not pass this randomized test; however, this approach does not obviously extend to estimating the confidence. Further, the space cost of this test is polynomial in the input size N and requires prior knowledge of N, whereas the solutions we seek are largely independent of this quantity, or depend only very weakly (logarithmically) on N. Estimating FDs from a sample was also considered in [20], but no error bounds were given and the confidence metric was different (the number of distinct antecedent values divided by the number of distinct antecedent-consequent values).

There has also been work on estimating other types of integrity constraints from a (random) sample, such as association rules [26, 9], algebraic relationships between attributes [7], and universal first order logic sentences [22]. An association rule $X \rightarrow Y$ over a set of transactions (e.g., $\{\texttt{bread}, \texttt{milk}\} \rightarrow \{\texttt{butter}\}$) asserts that whenever all the items in the set X appear in a transaction, so do all the items in the set Y. The confidence of an association rule is the count of transactions having X and Y divided by the number of transactions having X. Thus, approximating the confidence of an association rule requires an estimate of only these two counts, which can be obtained from a random sample with additive error. In contrast, CFD estimation involves counting the most frequent consequent value for each antecedent group. Similarly, [7] discovers correlations between attributes by estimating frequency counts of attribute values from a random sample, and [22] estimates the confidence of logical sentences by counting the proportion of tuples in the random sample that satisfy them.

Our work is related to summarizing large data sets. There has been recent work on maintaining bounded-size samples in a data warehouse that continually receives new data, but only random samples were considered [8, 17]. Our (one-pass and two-pass) algorithms can be thought of as constructing special summaries that may be stored in a data warehouse for CFD confidence estimation. These summaries are formed by careful combination of existing summaries such as reservoir sampling [27], heavy-hitter summaries [25], and count-min sketches [13]. Lastly, the CFD confidence estimation problem can be thought of as a special case of a "cascaded aggregate" [12], formed by the summation of frequent items. In [12], this problem is denoted as $F_1(F_\infty)$, for which solutions were not previously known. Similar multilevel techniques have been previously used to estimate other quantities over data streams [2].

9. CONCLUDING REMARKS

We have introduced the problem of estimating the confidence of Conditional Functional Dependencies when the tableau is given upfront or specified after the fact. It is natural to extend our results to related problems; we briefly outline some extensions to which they apply. Given a tableau, it is sometimes important to know not just the *global* confidence of the CFD, but also the *local* confidence resulting from each row in isolation [18]. However, this can be thought of as a generalization of the variable CFD problem: each row of the tableau can be thought of as an entire tableau, and the relevant confidence estimated.

Given a CFD with less than perfect confidence, it is also informative to know for which subset of the data the CFD *fails* to hold. This is captured by a *fail tableau*, by analogy with the (hold) tableau on which the CFD holds with high confidence [18]. Given a fail tableau, it is straightforward for our methods to estimate its (low) confidence. Similarly, our algorithms apply to extended CFDs introduced in [3], whose tableaux may include negations and disjunctions in addition to constants and wildcards. A last challenging problem is to summarize the data, and then to attempt to discover a high confidence tableau for a particular embedded FD from the summary [18]. A natural direction to study is to apply known tableau discovery algorithms directly to the data stored in the summaries we have defined here, and also use the summary to estimate the confidence of the candidate tableau rows. We hope to report on the quality of this approach in future work.

10. REFERENCES

[1] M. Arlitt and T. Jin. 1998 world cup web site access logs. http://www.acm.org/sigcomm/ITA/, 1998.

[2] L. Bhuvanagiri, S. Ganguly, D. Kesh, and C. Saha. Simpler algorithm for estimating frequency moments of data streams. In *ACM-SIAM Symposium on Discrete Algorithms*, 2006.

[3] L. Bravo, W. Fan, F. Geerts, and S. Ma. Increasing the expressivity of conditional functional dependencies without extra complexity. In *IEEE International Conference on Data Engineering*, 2008.

[4] L. Bravo, W. Fan, and S. Ma. Extending dependencies with conditions. In *International Conference on Very Large Data Bases*, 2007.

[5] A. Broder, M. Charikar, A. Frieze, and M. Mitzenmacher. Min-wise independent permutations. In *ACM Symposium on Theory of Computing*, 1998.

[6] Joshua Brody and Amit Chakrabarti. A multi-round communication lower bound for gap hamming and some consequences. *CoRR*, abs/0902.2399, 2009.

[7] P. Brown and P. Haas. BHUNT: Automatic discovery of fuzzy algebraic constraints in relational data. In *International Conference on Very Large Data Bases*, 2003.

[8] P. Brown and P. Haas. Techniques for warehousing of sample data. In *IEEE International Conference on Data Engineering*, 2006.

[9] B. Chen, P. Haas, and P. Scheuermann. A new two-phase sampling based algorithm for discovering association rules. In *ACM SIGKDD*, 2002.

[10] F. Chiang and R. Miller. Discovering data quality rules. In *International Conference on Very Large Data Bases*, 2008.

[11] G. Cong, W. Fan, F. Geerts, X. Jia, and S. Ma. Improving data quality: Consistency and accuracy. In *International Conference on Very Large Data Bases*, 2007.

[12] G. Cormode and S. Muthukrishnan. Space efficient mining of multigraph streams. In *ACM Principles of Database Systems*, 2005.

[13] G. Cormode and S. Muthukrishnan. Summarizing and mining skewed data streams. In *SIAM Conference on Data Mining*, 2005.

[14] W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Conditional functional dependencies for capturing data inconsistencies. *ACM Trans. Database Syst.*, 33(2), 2008.

[15] W. Fan, F. Geerts, L. Lakshmanan, and M. Xiong. Discovering conditional functional dependencies. In *IEEE International Conference on Data Engineering*, 2009.

[16] W. Fan, S. Ma, Y. Hu, J. Liu, and Y. Wu. Propagating functional dependencies with conditions. In *International Conference on Very Large Data Bases*, 2008.

[17] R. Gemulla, W. Lehner, and P. Haas. A dip in the reservoir: Maintaining sample synopses of evolving datasets. In *International Conference on Very Large Data Bases*, 2006.

[18] Lukasz Golab, Howard Karloff, Flip Korn, Divesh Srivastava, and Bei Yu. On generating near-optimal tableaux for conditional functional dependencies. In *International Conference on Very Large Data Bases*, 2008.

[19] Y. Huhtala, J. Karkkainen, P. Porkka, and H. Toivonen. TANE: An efficient algorithm for discovering functional and approximate dependencies. *The Computer Journal*, 42(2):100–111, 1999.

[20] I. Ilyas, V. Markl, P. Haas, P. Brown, and A. Aboulnaga. CORDS: Automatic discovery of correlations and soft functional dependencies. In *ACM SIGMOD International Conference on Management of Data*, 2004.

[21] T. S. Jayram, Ravi Kumar, and D. Sivakumar. The one-way communication complexity of hamming distance. *Theory of Computing*, 4(1):129–135, 2008.

[22] J. Kivenen and H. Mannila. The power of sampling in knowledge discovery. In *ACM Principles of Database Systems*, 1994.

[23] J. Kivinen and H. Mannila. Approximate inference of functional dependencies from relations. *Theor. Comput. Sci.*, 149(1), 1995.

[24] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[25] A. Metwally, D. Agrawal, and A. El Abbadi. Efficient computation of frequent and top-k elements in data streams. In *International Conference on Database Theory*, 2005.

[26] H. Toivonen. Sampling large databases for association rules. In *International Conference on Very Large Data Bases*, 1996.

[27] J. S. Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, 11(1):37–57, March 1985.