Analysis and Application of Adaptive Sampling

James F. Lynch*
Department of Mathematics and Computer Science
Box 5815
Clarkson University
Potsdam NY 13699-5815
jlynch@clarkson.edu

ABSTRACT

An estimation algorithm for a query is a probabilistic algorithm that computes an approximation for the size (number of tuples) of the query. The main question that is studied is which classes of logically definable queries have fast estimation algorithms. Evidence from descriptive complexity theory is provided that indicates not all such queries have fast estimation algorithms. However, it is shown that on classes of structures of bounded degree, all first-order queries have fast estimation algorithms.

These estimation algorithms use a form of statistical sampling known as adaptive sampling. Several versions of adaptive sampling have been developed by other researchers. The original version has been surpassed in some ways by a newer version and a more specialized Monte-Carlo algorithm. An analysis of the average runtime of the original version is given, and the different algorithms are compared. The analysis is used to compute what appears to be the best known upper bound on the efficiency of the original algorithm. Also, contrary to what seems to be a commonly held opinion, the two methods of adaptive sampling are incomparable. Which method is superior depends on the query being estimated and the criteria that are being applied. Lastly, adaptive sampling can be more efficient than the Monte-Carlo algorithm if knowledge about the maximum values of the data being sampled is available.

Categories and Subject Descriptors

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

General Terms

Queries, Sampling

1. INTRODUCTION

*Research Supported by NSF Grant CCR-9406809

Estimating the size of a query can be described as the following problem. Given a database $\mathfrak{A}$, let R be the query evaluated on $\mathfrak{A}$. $|R|$ is the size of R, i.e., the number of tuples in it. Let A be a probabilistic algorithm such that when given $\mathfrak{A}$, it returns an estimate $\mathbf{X}$ for $|R|$. Suppose that there is $E > 0$ and a probability $p < 1$ such that

$$\mathrm{pr}(|\mathbf{X} - |R|| \leq E) \geq p \ .$$

Then we say that A estimates $|R|$ with error E and confidence p. We are interested in finding fast algorithms that can estimate $|R|$ to any desired degree of accuracy and confidence. "Fast" can be interpreted in two ways. Either the estimation algorithm runs in time $o(t(n))$, where $t(n)$ is the time required to compute $|R|$ exactly, or its running time has low complexity, say linear or even constant.

There are several reasons why database researchers are interested in fast algorithms for estimating the size of queries. Some queries, in particular those defined recursively, can be quite expensive to compute. If the only important question is their size, then a fast estimation algorithm has significant practical value. Even if an exact computation of the query is needed, a fast estimation of its size can help to decide whether it is feasable to actually perform the computation. Also, the speed of computing the composition of simpler queries can depend on the order of their execution. Estimations of their size can help determine the optimal order of execution.

An obvious question is whether such algorithms exist for all queries. The following argument based on descriptive complexity theory [7] indicates that this is unlikely. For any natural number n, any $s \in \{0,1\}^n$ can be associated with a finite structure $\mathfrak{A}_s$ whose universe is $\{1,\ldots,n\}$. For any PSPACE language $L \subseteq \{0,1\}^*$, there is a formula $\phi(x)$ in partial fixed point logic such that for all s, it defines a unary query $R = \{i \in \{1,\ldots,n\} \colon \mathfrak{A}_s \models \phi(i)\}$ such that $|R| = 0$ or $|R| = n$, and $s \in L$ if and only if $|R| = n$. Taking $E = |R|/3$, if there were fast estimation algorithms for ϕ, then for every probability $p < 1$, there would be a fast probabilistic decision algorithm A such that for all $s \in \{0,1\}^*$, with probability at least p, A accepts s if and only if $s \in L$.

This does not preclude the possibility that there are large classes of databases and queries that do have fast estimation algorithms, or whose average runtime is fast. In particular, do there exist fast estimation algorithms for all first-order

queries on all databases? We do not know the answer to this question, but an article by Abiteboul, Compton, and Vianu [1] proposed the class of databases whose relations have bounded degree as an important class to investigate. They were not studying estimation algorithms; rather, they considered probabilistic methods for speeding up the average time for exact evaluation of higher-order queries. We will show that there are fast average time algorithms for estimating all first-order queries on databases of bounded degree.

There are actually two notions of average time complexity for database algorithms. The first regards the database as fixed, and the algorithm makes random choices in computing its estimate. The second assumes the database itself is random. (The algorithm may or may not be probabilistic.) In this article, we concentrate on the first notion. Specifically, we investigate a statistical sampling method for estimating the size of queries in databases.

A variety of methods for sampling queries in databases have been proposed. One approach is a form of sequential sampling. It relies on a partitioning of the query into sets that agree on the value of some attribute. More abstractly, let n be a natural number and $R \subseteq \{1, \ldots, n\}^r$ be an r-ary relation, i.e., relational query. For $i = 1, \ldots, n$, let $R_i = \{(i_2, \ldots, i_r) : (i, i_2, \ldots, i_r) \in \{1, \ldots, n\}^r\}$. Then $|R| = \sum_{i=1}^{n} |R_i|$. The sampling algorithm repeatedly makes random choices of $i \in \{1, \ldots, n\}$, calculates $|R_i|$, and adds it to a cumulative sum until a predetermined stopping condition is satisfied. Then it produces an estimate of $|R|$.

Statistical inference based on the sum of a sequence of random variables was used by W. Bartky [2] for other applications, and the general theory of sequential sampling was formulated by A. Wald [19, 20]. These authors considered the more general case when the sum could have negative terms. R. Lipton and J. Naughton [16] developed a version of sequential sampling which estimated the size of queries as outlined above. Their algorithm stopped sampling when the sum exceeded a predetermined value. They called it adaptive sampling because the number of samples taken was not predetermined; it depended on how quickly the sum reached the upper bound. Actually, similar stopping conditions had been used by Bartky and Wald. In their case, the sampling stopped when the sum either exceed an upper bound or fell below a lower bound.

The upper bound used by Lipton and Naughton was of the form $cb(n)$, where $b(n)$ is an *a priori* upper bound on the size of the partitions. (Thus, with no additional knowledge, $b(n) = n^{r-1}$.) Their error term was of the form $\epsilon|R|$, for some constant $\epsilon > 0$. They claimed that, by making c large enough, any desired degree of accuracy can be obtained with arbitrarily high probability. That is, for any $\epsilon > 0$ and $p \in [0, 1)$, the probability that the estimate is within $\epsilon|R|$ of $|R|$ can be made as large as p. This is true, but the proof for their bound on c is incorrect. We present a very different proof with a slightly larger bound on c.

A later version of adaptive sampling is due to P. Haas and A. Swami [11]. It does not make any *a priori* assumptions about the data except that not all the $|R_i|$ are equal. Haas and Swami prove that, for fixed R, as $\epsilon \to 0$, the probability that their method produces an estimate that is within $\epsilon|R|$ of $|R|$ approaches p. Further, their algorithm is asymptotically efficient in the sense that as $\epsilon \to 0$, the number of samples taken is close to the minimum number needed to get an estimate that is within $\epsilon|R|$ of $|R|$ with probability p. Thus their algorithm will perform better than the Lipton-Naughton algorithm on fixed samples as the error factor ϵ goes to 0. Although Haas and Swami clearly state in their paper that their estimate may not be accurate if ϵ does not get small, it appears to be the belief of some database researchers that the Haas-Swami algorithm is generally better than the Lipton-Naughton algorithm. But the two algorithms are actually incomparable. We will show that, for fixed ϵ and highly skewed data, the Lipton-Naughton algorithm can give a better estimate than the Haas-Swami algorithm.

2. ADAPTIVE SAMPLING

Adaptive sampling is a general method that applies to estimating the sum of a finite sequence of nonnegative integers. For a natural number n, let $a_1, \ldots, a_n$ be a sequence of integers such that $a_i \le b(n)$ for $i = 1, \ldots, n$, where b is a function of n. The intention is that $a_i = |R_i|$, $b(n) = n^{r-1}$, and $\sum_{i=1}^{n} a_i = |R|$. In [16], R. Lipton and J. Naughton introduced the following algorithm for estimating $A = \sum_{i=1}^{n} a_i$. The c is a fixed constant independent of n.

ALGORITHM 1.

$S \leftarrow 0$
$m \leftarrow 0$
repeat
  do $j \leftarrow$ RANDOM$(1, n)$
    $S \leftarrow S + a_j$
    $m \leftarrow m + 1$
until $S \ge cb(n)$
return nS/m

In a later article [17], the same authors claimed that any degree of accuracy and confidence could be obtained from this algorithm by taking c large enough. Specifically,

CLAIM 1. *Assume $a_i > 0$ for $i = 1, \ldots, n$. For $0 \le p < 1$ and $d > 0$, if $c = d(d+1)/(1-\sqrt{p})$, then*

$$\mathrm{pr}(|nS/m - A| \le A/d) \ge p \ .$$

Lipton and Naughton used an urn model to show that, by taking a sufficiently large constant α, the probability that the algorithm stops before $\alpha b(n)$ steps can be made arbitrarily small, and for any given $m \ge \alpha b(n)$, $\mathrm{pr}(|nS/m - A| > A/d)$ can also be made arbitrarily small. However, this does not imply their Claim because they still need to show either

1. for any $m \ge \alpha b(n)$, the conditional probability that $|nS/m - A| > A/d$, given that the algorithm stops at m repetitions, is arbitrarily small, or
2. the sum over all $m \ge \alpha b(n)$ of $\mathrm{pr}(|nS/m - A| > A/d)$ is arbitrarily small.

Method 1. seems quite involved, while method 2. requires a very tight bound on the probabilities being summed. Obvious approaches are suggested by the Central Limit Theorem, but as we will show, the error term in the Central Limit Theorem is so large that such a tight bound cannot be obtained. Similar problems arise in attempting to use Chernoff [5, 12] or Hoeffding [15] bounds.

Using a very different approach, we will analyze the average running time of Algorithm 1. Our results will enable us to derive an upper bound on c that is almost as good as that in the Claim. We will model the Algorithm as a 1-dimensional random walk, where at each step, the particle makes a jump to the right, the distance being randomly chosen from $\{a_1, \ldots, a_n\}$. Here, the particle's initial position is 0, and the final m is the time at which the particle passes through an absorbing barrier located at $cb(n)$. With this interpretation, for $0 \leq p < 1$ and $\epsilon > 0$, we will show how to compute c such that for all n, with probability greater than or equal to p, m will be within $\epsilon cnb(n)/A$ of $cnb(n)/A$. It follows that, by taking c sufficiently large, any desired degree of accuracy and confidence can be obtained. Our bound on c is not as good as that in the Claim. However, the condition that $a_i > 0$ for all $i = 1, \ldots, n$ can be relaxed. We need to assume only that $a_i \geq 0$ for all such i, and $a_i > 0$ for at least one i.

Consider a particle that performs a 1-dimensional random walk at discrete time steps, with an absorbing barrier at $cb(n)$. Its position at time 0 is 0, and at any given time m, its location is some integer k. It will continue its walk if $k < cb(n)$, in which case its position at time $m+1$ will be $k + a_i$, for some randomly chosen $i \in \{1, \ldots, n\}$, where all choices are equally likely. Thus if the particle is located at 0 at time 0, its location at each time m is the value of S in the Algorithm after m iterations of the **repeat** ... **until** loop, and the time until absorption is the final value of m. The theorem will be a consequence of a probabilistic estimation of the absorption time.

For each $k \geq 0$, let $\mathbf{T}_k$ be the random variable that is the absorption time of the particle, when initially located at k. We will derive recurrences for $\mathbf{E}[\mathbf{T}_k]$ and $\mathbf{Var}[\mathbf{T}_k]$. For $k, m \geq 0$, let $p_{k,m}$ be the probability that $\mathbf{T}_k = m$, and let $p_k(z) = \sum_{m=0}^{\infty} p_{k,m} z^m$ be the generating function of $\langle p_{k,m} : m \geq 0 \rangle$. Then

$$\begin{aligned} \mathrm{pr}(\mathbf{T}_k < \infty) &= p_k(1) \ , \\ \mathbf{E}[\mathbf{T}_k] &= p_k'(1) \text{ and} \\ \mathbf{Var}[\mathbf{T}_k] &= p_k''(1) + p_k'(1) - p_k'(1)^2 \ . \end{aligned}$$

It can be seen that the $p_{k,m}$ satisfy the following recurrence:

$$\begin{aligned} p_{k,0} &= 1 \text{ for } k \geq cb(n) \\ p_{k,0} &= 0 \text{ for } k < cb(n) \\ p_{k,m} &= 0 \text{ for } k \geq cb(n) \text{ and } m > 0 \\ p_{k,m+1} &= \sum_{i=1}^{n} p_{k+a_i,m}/n \text{ for } k < cb(n) \ . \end{aligned}$$

Therefore, for $k < cb(n)$,

$$p_k(z) = z \sum_{i=1}^{n} p_{k+a_i}(z)/n \ , \tag{1}$$

$$p_k'(z) = \sum_{i=1}^{n} [p_{k+a_i}(z) + z p_{k+a_i}'(z)]/n \ , \text{ and} \tag{2}$$

$$p_k''(z) = \sum_{i=1}^{n} [2p_{k+a_i}'(z) + z p_{k+a_i}''(z)]/n \ . \tag{3}$$

In the following, $H = \{i : a_i > 0\}$ and $h = |H|$.

LEMMA 2. *For all integers k,*

$$p_k(1) = 1 \ .$$

PROOF. This follows from more general theorems about unlimited Bernoulli trials or random walks (see e.g. Feller [8]), but we give an elementary proof. The Lemma is obvious for $k \geq cb(n)$. We will prove it for the remaining values by decreasing induction on k.

Thus, assume $k < cb(n)$ and the Lemma holds for all $k' > k$. Recalling our assumption that $h > 0$, by (1) and the induction assumption,

$$p_k(1) = \frac{h}{n} + \left(\frac{n-h}{n}\right) p_k(1) \ ,$$

and the Lemma follows. □

LEMMA 3. *For $k \leq (c+1)b(n)$,*

$$\frac{n(cb(n) - k)}{A} \leq p_k'(1) \leq \frac{n((c+1)b(n) - k)}{A} \ .$$

PROOF. We use decreasing induction on $k = (c+1)b(n)$ down to 0. For $cb(n) \leq k \leq (c+1)b(n)$, $p_k(z) = 1$, and the result is obvious.

Now assume $k < cb(n)$ and the result holds for all k' such that $k < k' \leq (c+1)b(n)$. By (2) and Lemma 2,

$$\begin{aligned} p_k'(1) &= \sum_{i=1}^{n} [1 + p_{k+a_i}'(1)]/n \ , \text{ and} \\ \left(\frac{h}{n}\right) p_k'(1) &= 1 + \sum_{i \in H} p_{k+a_i}'(1)/n \\ &\leq 1 + \sum_{i \in H} \frac{(c+1)b(n) - k - a_i}{A} \\ &\quad \text{by the induction assumption} \\ &= \frac{h((c+1)b(n) - k)}{A} \ . \end{aligned}$$

The lower bound is similar. □

LEMMA 4. *For $k \le (c+1)b(n)$,*

$$\begin{aligned} p_k''(1) &\le \frac{n^2((c+1)b(n)-k)^2}{A^2} \\ &\quad + \frac{n^2(\sum_{i=1}^n a_i^2)((c+1)b(n)-k)}{A^3} \\ &\quad - \frac{2n(cb(n)-k)}{A} \ . \end{aligned}$$

PROOF. We again use decreasing induction on k, and the result is obvious for $cb(n) \le k \le (c+1)b(n)$.

Now assume $k < cb(n)$ and the inequality holds for all k' such that $k < k' \le (c+1)b(n)$. By (3), Lemma 3, and the induction assumption,

$$\begin{aligned} hp_k''(1) &\le \sum_{i=1}^n \frac{2n((c+1)b(n)-k-a_i)}{A} \\ &\quad + \sum_{i \in H} \left[\frac{n^2((c+1)b(n)-k-a_i)^2}{A^2} \right. \\ &\qquad + \frac{n^2(\sum_{i=1}^n a_i^2)((c+1)b(n)-k-a_i)}{A^3} \\ &\qquad \left. - \frac{2n(cb(n)-k-a_i)}{A} \right] \\ &= \frac{2n^2((c+1)b(n)-k)}{A} - \frac{2n\sum_{i=1}^n a_i}{A} \\ &\quad + \frac{hn^2((c+1)b(n)-k)^2}{A^2} + \frac{n^2\sum_{i=1}^n a_i^2}{A^2} \\ &\quad - \frac{2n^2((c+1)b(n)-k)\sum_{i=1}^n a_i}{A^2} \\ &\quad + \frac{hn^2(\sum_{i=1}^n a_i^2)((c+1)b(n)-k)}{A^3} \\ &\quad - \frac{n^2(\sum_{i=1}^n a_i^2)\sum_{i=1}^n a_i}{A^3} \\ &\quad - \frac{2hn(cb(n)-k)}{A} + \frac{2n\sum_{i=1}^n a_i}{A} \\ &= \frac{hn^2((c+1)b(n)-k)^2}{A^2} \\ &\quad + \frac{hn^2(\sum_{i=1}^n a_i^2)((c+1)b(n)-k)}{A^3} \\ &\quad - \frac{2hn(cb(n)-k)}{A} \ . \end{aligned}$$

□

From Lemmas 2, 3, and 4, taking $m = \mathbf{T}_0$,

THEOREM 5.

$$\begin{aligned} \mathrm{pr}(m < \infty) &= 1 \ , \\ \frac{cnb(n)}{A} \le \mathbf{E}[m] &\le \frac{(c+1)nb(n)}{A} \text{ and} \\ \mathbf{Var}[m] &\le \frac{(2c+1)n^2b(n)^2}{A^2} \\ &\quad + \frac{(\sum_{i=1}^n a_i^2)(c+1)n^2b(n)}{A^3} + \\ &\quad \frac{(1-c)nb(n)}{A} \ . \end{aligned}$$

We will now use these formulae to compute an upper bound on c in Algorithm 1.

THEOREM 6. *For $0 \le p < 1$ and $\epsilon > 0$, there exists c such that for all n,*

$$\mathrm{pr}(|nS/m - A| \le \epsilon A) \ge p \ .$$

PROOF. By Chebyshev's inequality, for any $\gamma > 0$,

$$\begin{aligned} \mathrm{pr}\left(|m - \mathbf{E}[m]| \ge \frac{\gamma cnb(n)}{A} \right) &\le \frac{\mathbf{Var}[m]A^2}{\gamma^2c^2n^2b(n)^2} \\ &\le \frac{2c+1}{\gamma^2c^2} \\ &\quad + \frac{(c+1)\sum_{i=1}^n a_i^2}{\gamma^2c^2b(n)A} + \\ &\quad \frac{(1-c)A}{\gamma^2c^2nb(n)} \ . \end{aligned}$$

Since $\sum_{i=1}^n a_i^2 \le b(n)A$ and $c > 1$,

$$\begin{aligned} \mathrm{pr}\left(|m - \mathbf{E}[m]| \ge \frac{\gamma cnb(n)}{A} \right) &\le \frac{2c+1}{\gamma^2c^2} + \frac{c+1}{\gamma^2c^2} \\ &= \frac{3c+2}{\gamma^2c^2} \ . \end{aligned} \tag{4}$$

Since

$$\begin{aligned} \left| m - \frac{cnb(n)}{A} \right| &\le |m - \mathbf{E}[m]| + \left| \mathbf{E}[m] - \frac{cnb(n)}{A} \right| \\ &\le |m - \mathbf{E}[m]| + \frac{nb(n)}{A} \text{ by Lemma 3,} \end{aligned}$$

in order to guarantee that

$$\mathrm{pr}\left(\left| m - \frac{cnb(n)}{A} \right| \le \frac{\gamma cnb(n)}{A} \right) \ge p \ ,$$

it suffices that

$$\mathrm{pr}\left(|m - \mathbf{E}[m]| \le \frac{\gamma cnb(n)}{A} - \frac{nb(n)}{A} \right) \ge p \ .$$

Therefore, by (4), it suffices to take c and γ such that

$$\gamma c - 1 > 0 \text{ and} \tag{5}$$

$$\frac{3c+2}{(\gamma c-1)^2} \le 1 - p \ . \tag{6}$$

With probability 1, $cb(n) \le S < (c+1)b(n)$. Therefore if

$$\left| m - \frac{cnb(n)}{A} \right| \le \frac{\gamma cnb(n)}{A} \ ,$$

then, assuming $\gamma < 1$,

$$\begin{aligned} \frac{cnb(n)A}{(1+\gamma)cnb(n)} \le \frac{nS}{m} &\le \frac{(c+1)nb(n)A}{(1-\gamma)cnb(n)}, \text{ i.e.,} \\ \frac{A}{1+\gamma} \le \frac{nS}{m} &\le \frac{(c+1)A}{(1-\gamma)c} \ . \end{aligned}$$

Since we want $|nS/m - A| \le \epsilon A$, it suffices to take

$$\begin{aligned} 1 - \epsilon \le \frac{1}{1+\gamma} \quad &\text{and} \quad \frac{c+1}{(1-\gamma)c} \le 1 + \epsilon, \text{ or} \\ \gamma \le \frac{\epsilon}{1-\epsilon} \quad &\text{and} \quad \frac{1}{c} \le \epsilon - \gamma - \gamma\epsilon \ . \end{aligned}$$

(The first condition above applies when $\epsilon < 1$. If $\epsilon \geq 1$, then it can be omitted.) In order that $c > 0$, the second of the above conditions requires that $\gamma < \epsilon/(1+\epsilon)$, which is a stronger condition than the first. Therefore it suffices to take

$$\gamma < \frac{\epsilon}{1+\epsilon} \text{ and} \quad (7)$$

$$c \geq \frac{1}{\epsilon - \gamma - \gamma\epsilon} \ . \quad (8)$$

All four conditions (5)–(8) can be satisfied by taking $0 < \gamma < \epsilon/(1+\epsilon)$ and $c > \max(1/\gamma, 1/(\epsilon - \gamma - \gamma\epsilon), \rho)$, where ρ is the larger root of

$$\frac{3x+2}{(\gamma x - 1)^2} = 1 - p \ ,$$

i.e.,

$$\rho = \frac{1}{\gamma} + \frac{3 + \sqrt{8(1-p)\gamma^2 + 12(1-p)\gamma + 9}}{2(1-p)\gamma^2} \ .$$

It can be seen that $1/\gamma \leq \rho$, so we can take

$$c > \min_{\gamma \in (0, \epsilon/(1+\epsilon))} \max(1/(\epsilon - \gamma - \gamma\epsilon), \rho) \ . \quad (9)$$

□

The expression in (9) is rather cumbersome, but it has some simpler approximations. As $\epsilon \to 0$,

$$\rho \to \frac{1}{\gamma} + \frac{\gamma+3}{(1-p)\gamma^2} \leq \frac{\gamma+3}{(1-p)\gamma^2} \ .$$

It can be seen that over the interval $(0, \epsilon/(1+\epsilon))$, $(\gamma+3)/((1-p)\gamma^2)$ is decreasing and $1/(\epsilon - \gamma - \gamma\epsilon)$ is increasing, and the functions are asymptotic to infinity at opposite ends of this interval. Therefore there is a unique $\gamma \in (0, \epsilon/(1+\epsilon))$ where these functions are equal, and their value at this point is greater than the value of the expression in (9).

Solving

$$\frac{\gamma+3}{(1-p)\gamma^2} = \frac{1}{\epsilon - \gamma - \gamma\epsilon}$$

for γ,

$$\gamma = \frac{-3 - 2\epsilon + \sqrt{(3+2\epsilon)^2 + 12(2+\epsilon-p)\epsilon}}{2(2+\epsilon-p)} \sim \epsilon \text{ as } \epsilon \to 0 \ .$$

Since $(\gamma+3)/((1-p)\gamma^2)$ is continuous in a neighborhood around $\epsilon/(1+\epsilon)$, c can be approximated by

$$\frac{3}{(1-p)\epsilon^2} \ .$$

The bound given in Claim 1 is asymptotic to $1/((1-\sqrt{p})\epsilon^2)$ as $\epsilon \to 0$ (i.e., $d \to \infty$), which is smaller than our bound. Since $1 - p = (1+\sqrt{p})(1-\sqrt{p}) \sim 2(1-\sqrt{p})$ as $p \to 1$, our bound is larger than the bound in Claim 1 by a factor of about 1.5 as $\epsilon \to 0$ and $p \to 1$.

3. COMPARISON TO OTHER SAMPLING ALGORITHMS

P. Haas and S. Swami [11] proposed a variation on Algorithm 1. They pointed out that the stopping condition $S \geq cb(n)$ can be overly conservative if $b(n)$ is actually much larger than $\max(a_1, \ldots, a_n)$. Their stopping condition is based on the Central Limit Theorem. Let S_m be the value of S after m iterations of the **repeat** ... **until** loop in Algorithm 1. Then, as m gets large,

$$\text{pr}\left(\left|\frac{nS_m}{m} - A\right| \leq \epsilon A\right) = \text{pr}\left(\left|\frac{S_m - m\mu}{\sqrt{m}\sigma}\right| \leq \frac{\sqrt{m}\epsilon\mu}{\sigma}\right) \approx 2\Phi\left(\frac{\sqrt{m}\epsilon\mu}{\sigma}\right) - 1 \ , \quad (10)$$

where $\mu = \sum_{i=1}^{n} a_i/n$ is the expectation of a randomly chosen a_i, $\sigma^2 = \sum_{i=1}^{n}[a_i - \mu]^2/n$ is its variance, and

$$\Phi(t) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\, dx$$

is the standard normal distribution function.

Let

$$\zeta_p = \Phi^{-1}((1+p)/2) \text{ and}$$
$$m = \frac{\zeta_p^2 \sigma^2}{\epsilon^2 \mu^2} \ ,$$

Then, as ϵ approaches 0, m gets arbitrarily large and the error term in (10) will go to 0. Therefore for sufficiently small ϵ, the estimate nS_m/m will be within ϵA of A with probability p.

Of course, μ and σ are not known *a priori*. The algorithm of Haas and Swami makes successive estimates of μ and σ as it samples $\{a_1, \ldots, a_n\}$. It can be shown that with high probability, the mth estimates μ_m and σ_m approach their true values, and when $\sigma_m > 0$ and $m > \zeta_p^2 \sigma_m^2/(\epsilon^2 \mu_m^2)$, the probability that the estimate nS_m/m has the desired accuracy is at least p. Further, their method is asymptotically efficient, meaning that as ϵ gets small, their algorithm takes just enough samples to guarantee the desired accuracy and confidence.

Note that Theorem 6 does not make any assumptions about the asymptotics of the parameters, while Haas and Swami assume that $\epsilon \to 0$. As they point out, if ϵ is fixed, then it is possible that the error term in (10) is quite large, and the probability that the estimate is within the desired accuracy range could be less than p. Specifically, A. Berry [3, 4] has shown that the error term is bounded by

$$\frac{4\sum_{i=1}^{n} |a_i - \mu|^3}{\sqrt{m}\sigma^3} \ ,$$

which can be quite large if we know only that $a_1, \ldots, a_n \leq n$. This is the case when the size of a binary relation on $\{1, \ldots, n\}$ is being estimated. The following example illustrates this possibility.

For $i = 1, \ldots, n$ let

$$a_i = \begin{cases} 0 & \text{if } i < n \text{ and } i \text{ even} \\ 1 & \text{if } i < n \text{ and } i \text{ odd} \\ n & \text{if } i = n \end{cases}$$

Then, with probability greater than $(1-1/n)^n \approx 1/e$, a_n is not chosen in the first n iterations of the **repeat** ... **until** loop. If this happens, then the estimates of both μ and σ will rapidly approach 1/2, which almost surely will result in a stopping time less than $\omega(n)$, for any function ω that increases monotonically to ∞. More specifically, if ζ_p is sufficiently large, i.e., p is sufficiently close to 1, or ϵ is sufficiently small, then with probability close to $1/e$, the algorithm will return an estimate close to $n/2$, i.e., the probability that it is within ϵA of A will not be p.

A very different method for estimating the sizes of relations is a Monte-Carlo algorithm due to E. Cohen [6]. It repeats the following process some predetermined number of times, say t. In each iteration $s = 1, \ldots, t$, it performs the following two steps.

1. It randomly assigns a ranking $r_s(i) \in [0, \infty)$ to each $i \in \{1, \ldots, n\}$.
2. Then, for each $i \in \{1, \ldots, n\}$, it computes $\mu_s(i) = \min(r_s(j) : j \in R_i)$.

After the tth iteration, it estimates each $|R_i|$. There are several ways of doing this. Let $\hat{S}_i$ be the estimate for $|R_i|$. Using the average of the t samples:

$$\hat{S}_i = \frac{t}{\sum_{s=1}^{t} \mu_s(i)} \ ,$$

or, letting $\tilde{\mu}(i)$ be the $\lfloor t(1-1/e) \rfloor$-smallest value in $\{\mu_s(i) : 1 \le s \le t\}$,

$$\hat{S}_i = \frac{1}{\tilde{\mu}(i)} \ .$$

The idea behind this approach is that there is a strong correlation between $\mu_s(i)$ and $|R_i|$. The smaller $\mu_s(i)$ is, the larger $|R_i|$ is likely to be. As t increases, the probability that $\hat{S}_i$ is within a desired accuracy also increases. Specifically,

$$\mathrm{pr}(|\hat{S}_i - |R_i|| \le \epsilon |R_i|) = e^{-\Omega(\epsilon t)}$$

for each $i = 1, \ldots, n$, and

$$\mathrm{pr}(|\sum_{i=1}^{n} \hat{S}_i - |R|| \le \epsilon |R|) = e^{-\Omega(\epsilon t)} \ .$$

The efficiency of this algorithm depends on the speed of computing all the $\mu_s(i)$. At present, no efficient method applicable to all queries is known. An important case where there is a fast algorithm is the transitive closure of a binary relation. Letting $G = \langle \{1, \ldots, n\}, E \rangle$ be a graph on n vertices with $m = |E|$ edges and R_i be the set of vertices reachable from i by a path in G, $R = \cup_{i=1}^{n} R_i$ is the transitive closure of E. Cohen shows how to estimate each $|R_i|$ and $|R|$ in time $O(n+m)$. Using adaptive sampling, Lipton and Naughton [17] show how to estimate $|R|$ in time $O(n\sqrt{m})$, and therefore Cohen's method is more efficient on general graphs.

However, if it is known that the sizes of the reachability sets are small, then adaptive sampling can estimate the transitive closure faster than the Monte-Carlo method. For example, suppose it is known that the reachability sets have size at most k, where $k^2 = o(n)$. The Monte-Carlo algorithm will run in time $\Omega(n)$, but if the adaptive sampling algorithm uses $b(n) = k$, then it will take only $O(k^2)$ time to reach the same level of accuracy and confidence.

4. ESTIMATING QUERIES

A natural application of adaptive sampling is the estimation of queries, i.e., relations, on a finite structure. That is, we partition the relation R into disjoint sets and estimate $|R|$ by computing the sizes of randomly chosen partitions. Although the partitioning is often determined by the value of the first coordinate of the tuples in the relation (as outlined in the Introduction), there is no reason to expect that this is the optimal partitioning in all cases. Let R be r-ary. For any $j = 1, \ldots, r$, R can be partitioned into n^j sets indexed by elements in $\{1, \ldots, n\}^j$. Given $(i_1, \ldots, i_j) \in \{1, \ldots, n\}^j$,

$$R_{i_1,\ldots,i_j} = \{(i_{j+1}, \ldots, i_r) : (i_1, \ldots, i_r) \in R\} \ .$$

(Note that when $j = r$, $R_{i_1,\ldots,i_r} = \emptyset$ or $\{\Lambda\}$ where Λ is the null sequence.) Then $|R|$ can be estimated by sampling $|R_{i_1,\ldots,i_j}|$ for random $(i_1, \ldots, i_j)$.

Replacing n in Theorem 5 by n^j and $b(n)$ by n^{r-j},

THEOREM 7. *Let the cost of computing $|R_{i_1,\ldots,i_j}|$ on structures of size n be $f_j(n)$. Then for any $p \in [0,1)$ and $\epsilon > 0$, the average time required by adaptive sampling to produce an estimate $\tilde{A}$ such that*

$$\mathrm{pr}(|\tilde{A} - |R|| \le \epsilon |R|) \ge p$$

is $O(n^r f_j(n)/|R|)$.

Thus the efficiency of adaptive sampling can depend on the choice of j. In general, the fastest way to compute $|R_{i_1,\ldots,i_j}|$ may be to examine each element $(i_{j+1}, \ldots, i_r) \in \{1, \ldots, n\}^{r-j}$, and decide whether $(i_1, \ldots, i_r) \in R$. Letting $g(n)$ be the cost of computing this decision, the average time to estimate $|R|$ becomes $O(n^{2r-j} g(n)/|R|)$, which is minimal for $j = r$, i.e., $O(n^r g(n)/|R|)$. Since $|R|$ can be computed exactly in time $O(n^r g(n))$, adaptive sampling can be significantly faster if $|R|$ is large. However, the knowledge that $|R|$ is small relative to n^r can be useful, and it can be obtained by estimating $|\overline{R}|$, where $\overline{R}$ is the complement of R. A modified version of adaptive sampling can estimate $|\overline{R}|$ in parallel with the estimate of $|R|$. For example, Algorithm 1 would be modified as follows:

ALGORITHM 8.

$S \leftarrow 0$
$T \leftarrow 0$
$m \leftarrow 0$
repeat
  do $(i_1, \ldots, i_j) \leftarrow$ RANDOM$(\{1, \ldots, n\}^j)$
    $S \leftarrow S + |R_{i_1,\ldots,i_j}|$
    $T \leftarrow T + n^{r-j} - |R_{i_1,\ldots,i_j}|$
    $m \leftarrow m + 1$
until $S \ge cn^{r-j}$ **or** $T \ge cn^{r-j}$
return nS/m, nT/m

THEOREM 9. *Let the cost of deciding whether* $(i_1, \ldots, i_r) \in R$ *on structures of size* n *be* $g(n)$. *Then for any* $p \in [0,1)$ *and* $\epsilon > 0$, *the average time required by Algorithm 8 to produce an estimates* $\tilde{A}$, $\tilde{B}$ *such that*

$$\mathrm{pr}(|\tilde{A} - |R|| \leq \epsilon |R| \text{ or } \mathrm{pr}(|\tilde{B} - |\overline{R}|| \leq \epsilon |\overline{R}|) \geq p$$

is $O(g(n))$.

4.1 First-Order Queries

The two theorems above are quite general. Here, we examine the special case when R is defined by a first-order formula. That is, let $\phi(x_1, \ldots, x_r)$ be a first-order formula, and for any structure $\mathfrak{A}$ in the class, let $R_\phi = \{(a_1, \ldots, a_r) \in \{1, \ldots, n\}^r : \mathfrak{A} \models \phi(a_1, \ldots, a_r)\}$. For arbitrary $\mathfrak{A}$, we do not know of a fast estimation algorithm any better than the ones described in the two theorems above. But we will show that all first-order queries have fast estimation algorithms on finite structures of bounded degree. They rely on characterizations of first-order properties in terms of local "neighborhoods" of elements, first used by H. Gaifman [10] and W. Hanf [13] on arbitrary structures. Every member $\mathfrak{A}$ of such a class has the form $\langle \{1, \ldots, n\}, P_1, \ldots, P_k \rangle$, where n is a natural number, and each $P_i \subseteq \{1, \ldots, n\}^{p_i}$ for some fixed p_i. The Gaifman graph $\mathcal{G}(\mathfrak{A})$ of $\mathfrak{A}$ is $\langle \{1, \ldots, n\}, E \rangle$, where

$$\begin{aligned} E \quad = \quad & \{(a,b) \in \{1, \ldots, n\} : \text{ there exists } i \in \{1, \ldots, k\} \\ & \text{and } (a_1, \ldots, a_{p_i}) \in P_i \\ & \text{such that } \{a, b\} \subseteq \{a_1, \ldots, a_{p_i}\}\} \ . \end{aligned}$$

Letting δ be the usual notion of distance in the undirected graph $\mathcal{G}(\mathfrak{A})$, for any natural number d and $a_1, \ldots, a_i \in \{1, \ldots, n\}$, the d-neighborhood of $a_1, \ldots, a_i$ in $\mathfrak{A}$ is

$$N_d(a_1, \ldots, a_i) = \cup_{j=1}^{i} \{b \in \{1, \ldots, n\} : \delta(a_j, b) \leq d\} \ .$$

It is easy to define the property $\delta(x,y) \leq d$ by a formula in the first-order language of $P_1, \ldots, P_k$. A formula is said to be d-local if its quantifiers are of the form $(\forall x \in N_d(y_1, \ldots, y_i))$ or $(\exists x \in N_d(y_1, \ldots, y_i))$. Of course, every d-local formula is equivalent to a first-order formula. Further,

THEOREM 10 (H. GAIFMAN [10]). *Every first-order formula is equivalent to a Boolean combination of* d*-local formulae for some* d, *and sentences of the form*

$$\exists x_1 \ldots x_i (\bigwedge_{j=1}^{i} \alpha_j(x_j) \wedge \bigwedge_{1 \leq h \neq j \leq i} \delta(x_h, x_j) > 2m) \qquad (11)$$

for some m *and* m*-local formulae* α_j.

Let us call sentences of the form (11) background sentences of ϕ. The intention is that in order to compute or estimate $|R_\phi|$, one first determines whether the background sentences are satisfied. Note that this can be decided in time $O(n)$. Then, having done that, $|R|$ can be estimated just by estimating the number of tuples satisfying the d-local formulae of ϕ.

A class of structures is said to be of bounded degree if there is some b such that for every structure in the class, its Gaifman graph has degree bounded by b. In such a class, for every i and d, there is an upper bound on the sizes of the d-neighborhoods of all i-tuples of elements in the members of the class, and therefore each d-neighborhood is completely characterized up to isomorphism by some first-order formula. In this case, a modified version of adaptive sampling as illustrated in Algorithm 8 can estimate first-order queries in constant average time.

THEOREM 11. *On classes of structures of bounded degree, for any* $p \in [0,1)$ *and* $\epsilon > 0$ *and any first-order local formula* ψ, *adaptive sampling can produce estimates* $\tilde{A}$ *of* R_ψ *and* $\tilde{B}$ *of* $\overline{R}_\psi$ *such that*

$$\mathrm{pr}(|\tilde{A} - |R_\psi|| \leq \epsilon |R_\psi| \text{ or } \mathrm{pr}(|\tilde{B} - |\overline{R}_\psi|| \leq \epsilon |\overline{R}_\psi|) \geq p$$

in constant average time.

PROOF. Let $\psi(x_1, \ldots, x_i)$ be a d-local formula of ϕ. It is characterized by a finite collection of disjoint finite structures: the components of its isomorphism type. Arbitrarily select an element from each of these components. Without loss of generality, say they correspond to $x_1, \ldots, x_j$. Then we partition R according to j as above. Since all of the neighborhoods of $i_1, \ldots, i_j$ are of bounded size, there exists a constant b such that $|R_{\psi, i_1, \ldots, i_j}| \leq b$, and therefore $R_{\psi, i_1, \ldots, i_j}$ can be computed in $O(1)$ time. Then, applying Theorem 9, $|R_\psi|$ and $|\overline{R}_\psi|$ can be estimated in constant average time. □

COROLLARY 12. *On classes of structures of bounded degree, for any* $p \in [0,1)$ *and* $\epsilon > 0$ *and any first-order formula* ϕ, *adaptive sampling can produce estimates* $\tilde{A}$ *of* R_ϕ *and* $\tilde{B}$ *of* $\overline{R}_\phi$ *such that*

$$\mathrm{pr}(|\tilde{A} - |R_\phi|| \leq \epsilon |R_\phi| \text{ or } \mathrm{pr}(|\tilde{B} - |\overline{R}_\phi|| \leq \epsilon |\overline{R}_\phi|) \geq p$$

in constant average time $O(n)$.

5. FUTURE WORK

The problem of whether all first-order queries on arbitrary structures have fast estimation algorithms has already been mentioned. There is also the question of whether queries in more powerful logics can be estimated rapidly. For many of these logics, including transitive closure logic, the locality properties described by Gaifman and Hanf fail. Estimating query sizes, even on structures of bounded degree appears difficult in these cases. For example, consider a structure on $\{1, \ldots, n\}$ whose relations include successor $\{(i, i+1) : 1 \leq i < n\}$. One application of transitive closure on successor can define $\leq$, and once this is done, other applications of transitive closure can define any NLOGSPACE property. Then, as was shown for NSPACE in the Introduction, if there were fast estimation algorithms for transitive closure queries, for every NLOGSPACE language L, there would be fast probabilistic algorithms that decide membership in L with arbitrarily high confidence. This still leaves open the possibility that a *single* application of transitive closure to a first-order formula could be estimated quickly on structures of bounded degree. One of the Lipton-Naughton papers [16] studied fast estimation algorithms for the usual transitive closure relation on a graph, which is a special case of this problem. It seems likely that this algorithm,

combined with Gaifman's Theorem 10, would extend to the more general case.

Another potential area of investigation would be to consider broader classes of structures and logics that still obey locality conditions similar to Gaifman's and Hanf's. One example comes from a recent article by M. Frick and M. Grohe [9]. They developed fast algorithms for evaluating first-order sentences on locally tree-decomposable graphs, which are a generalization of bounded degree graphs.

An article by L. Hella, L. Libkin, and J. Nurmonen [14] shows that notions of locality extend to logics augmented by counting quantifiers. Since we are concerned here with approximate counting, a natural generalization of our results would be to consider approximate analogues of counting quantifiers. Let $\psi_1(x, u_1, \ldots, u_k)$ and $\psi_2(y, v_1, \ldots, v_l)$ be formulae. The analogue of the Rescher quantifier would be $Q_{\mathcal{R},\epsilon}$, where, for any structure $\mathfrak{A}$ on $\{1, \ldots, n\}$ and $i_1, \ldots, i_k, j_1, \ldots, j_l \in \{1, \ldots, n\}$,

$$\mathfrak{A} \models Q_{\mathcal{R},\epsilon} xy(\psi(x, i_1, \ldots, i_k), \psi(y, j_1, \ldots, j_l))$$

if and only if

$$|\{b\colon \mathfrak{A} \models \psi_1(b, i_1, \ldots, i_k)| \leq (1+\epsilon)|\{c\colon \mathfrak{A} \models \psi_2(c, j_1, \ldots, j_l)|.$$

The analogue of the Härtig quantifier would be $Q_{\mathcal{H},\epsilon}$, where

$$\mathfrak{A} \models Q_{\mathcal{H},\epsilon} xy(\psi(x, i_1, \ldots, i_k), \psi(y, j_1, \ldots, j_l))$$

if and only if

$$\begin{aligned} &||\{b\colon \mathfrak{A} \models \psi_1(b, i_1, \ldots, i_k)\}| \\ &\quad - |\{c\colon \mathfrak{A} \models \psi_2(c, j_1, \ldots, j_l)\}|| \\ &\quad \leq \epsilon|\{b\colon \mathfrak{A} \models \psi_1(b, i_1, \ldots, i_k)\}| \ . \end{aligned}$$

6. ACKNOWLEDGEMENTS

I thank Dr. Qiming Liu [18] for many stimulating conversations which exposed me to this area, and Dr. Peter Haas of the IBM Almaden Research Center for several very helpful email responses.

7. REFERENCES

[1] S. Abiteboul, K. Compton, and V. Vianu. Queries are easier than you thought (probably). In *Proc. ACM Symp. on Principles of Database Systems*, pages 23–32, 1992.

[2] W. Bartky. Multiple sampling with constant probability. *Ann. Math. Statist.*, 14:363–377, 1943.

[3] A. Berry. The accuracy of the gaussian approximation to the sum of independent variates. *Trans. AMS*, 49:122–136, 1941.

[4] J. Blum and J. Rosenblat. *Probability and Statistics.* W. B. Saunders Company, Philadelphia, 1972.

[5] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Ann. Math. Statist*, 23:493–507, 1952.

[6] E. Cohen. Size-estimation framework with applications to transitive closure and reachability. *J. Comput. System Sci.*, 55:441–453, 1997.

[7] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory.* Springer-Verlag, New York, Heidelberg, 1995.

[8] W. Feller. *An Introduction to Probability Theory and Its Applications, vol. 1.* John Wiley & Sons, New York, 1968.

[9] M. Frick and M. Grohe. Deciding first-order properties of locally tree-decomposable graphs. In *Proc. 26th ICALP*, pages 105–135, 1999.

[10] H. Gaifman. On local and nonlocal properties. In *Logic Colloquium '81*, pages 105–135. J. Stern, ed., North Holland, 1982.

[11] P. Haas and A. Swami. Sequential sampling, procedures for query size estimation. *IBM Research Report*, RJ 9101 (80915), 1992.

[12] T. Hagerup and C. Rüb. A guided tour of chernoff bounds. *Inf. Processing Letters*, 33:305–308, 1989/90.

[13] W. Hanf. Model-theoretic methods in the study of elementary logic. In *The Theory of Models*, pages 132–145. J. Addison, L. Henkin, and A. Tarski, eds., North Holland, 1965.

[14] L. Hella, L. Libkin, and J. Nurmonen. Notions of locality and their logical characterizations over finite models. *J. Symbolic Logic*, 64:1751–1773, 1999.

[15] W. Hoeffding. Probability inequalities for sums of bounded random variables. *J. Am. Statist. Assoc.*, 58:13–30, 1963.

[16] R. J. Lipton and J. F. Naughton. Estimating the size of generalized transitive closures. In *Proc. Fifteenth Intl. Conf. on Very Large Databases*, pages 165–172, 1989.

[17] R. J. Lipton and J. F. Naughton. Query size estimation by adaptive sampling. *J. Comp. Sys. Sci.*, 51:18–25, 1995.

[18] Q. Liu. *On algorithms for database query size estimation.* Master's Thesis, Clarkson University, 1998.

[19] A. Wald. On cumulative sums of random variables. *Ann. Math. Statist.*, 15:283–296, 1944.

[20] A. Wald. *Sequential Analysis.* John Wiley & Sons, New York, 1947.