A Sampling-Based Approach to Information Recovery†

Junyi Xie,[1] Jun Yang,[2] Yuguo Chen,[3] Haixun Wang,[4] Philip S. Yu[5]

[1]*Oracle Corporation*
Redwood City, California, USA
junyi.xie@oracle.com

[2]*Department of Computer Science, Duke University*
Durham, North Carolina, USA
junyang@cs.duke.edu

[3]*Department of Statistics, University of Illinois at Urbana-Champaign*
Champaign, Illinois, USA
yuguo@uiuc.edu

[4,5]*IBM T. J. Watson Research Center*
Hawthorne, New York, USA
{haixun,psyu}@cs.duke.edu

***Abstract*— There has been a recent resurgence of interest in research on noisy and incomplete data. Many applications require information to be recovered from such data. Ideally, an approach for information recovery should have the following features. First, it should be able to incorporate prior knowledge about the data, even if such knowledge is in the form of complex distributions and constraints for which no close-form solutions exist. Second, it should be able to capture complex correlations and quantify the degree of uncertainty in the recovered data, and further support queries over such data. The database community has developed a number of approaches for information recovery, but none is general enough to offer all above features. To overcome the limitations, we take a significantly more general approach to information recovery based on sampling. We apply *sequential importance sampling*, a technique from statistics that works for complex distributions and dramatically outperforms naive sampling when data is constrained. We illustrate the generality and efficiency of this approach in two application scenarios: cleansing RFID data, and recovering information from published data that has been summarized and randomized for privacy.**

I. Introduction

Recent database research has witnessed rising interests in managing incomplete and uncertain data. For some applications, such as sensor and RFID (Radio Frequency Identification) networks, data collection is inherently unreliable, resulting in errors and missing data [1]. In some scenarios, such as privacy-preserving data publishing, data is intentionally summarized and perturbed before being released for public use [2]. Even in traditional database systems, we find examples of incomplete data such as feedbacks from query execution, which provide summary information for portions of the database and can be used to infer database statistics for query optimization [3]. In all scenarios above, we are interested in recovering meaningful information from noisy, incomplete, and/or summarized data. In the following, we describe in more detail two applications, which will serve as running examples in this paper.

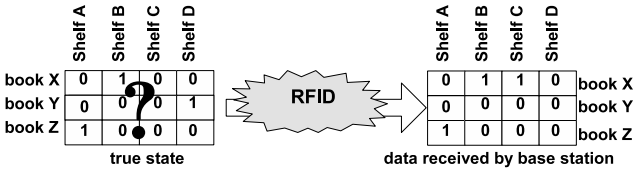

Fig. 1. Unreliable RFID data.

- *Information recovery from RFID readings (IR-RFID).* Consider a scenario where we deploy RFID detectors on library shelves to track locations of books [4]. Periodically, the detector on each shelf reports all books detected on that shelf and transmit the readings back to a base station, which collects all reports into a table shown on the right of Figure 1. A value of 1 (or 0) for table entry (i, j) means that book i is detected (or undetected, respectively) on shelf j in the current timestep. RFID data acquisition and transmission are unreliable; it is not uncommon for 30% of the readings to be dropped [1]. Hence, a number of anomalies may occur in this table. For example, book X may be detected simultaneously on two shelves, but it cannot possibly be on both shelves; book Y is detected on none of the shelves, even though it may be actually on shelf D. Therefore, we cannot interpret the data received by the base station as the true state of book locations; instead, we should try to recover the true state from the received data.
- *Information recovery for privacy-preserving data publishing (IR-PPDP).* In privacy-preserving data publishing, publishers usually summarize and randomly perturb data before making it available. Instead of releasing the

†The work was performed while the first author was at Duke University. The first and second authors were supported by an NSF CAREER award (IIS-0238386).

original table, which may be a commercial secret, the publisher releases a perturbed version where each table entry (sales of a make in a region) has been added a random noise. Meanwhile, government and market research agencies may also release summaries of the same data, e.g., total sales by make (with all region combined) and total sales by region (with all makes combined). Using the perturbed and summarized data, data analysts will try to recover as much information about the original table as possible. Data publishers may also be interested in this recovery procedure, in order to assess the extent of information disclosure.

We argue that an information recovery procedure should be able to incorporate prior knowledge about the data of interest, and be able to capture complex correlations and quantify the degree of uncertainty in the recovered data. In the following, we elaborate on these features with our two running examples.

- *Incorporation of constraints.* In many cases we may know some constraints on the data to be recovered, and it is natural to use such constraints to resolve inconsistency and reduce uncertainty. For example, in IR-RFID, the number of books on a shelf should not exceed the maximum capacity of the shelf; each book can be on at most one shelf at any time; and if a book is checked out, it should not be on any shelf. In IR-PPDP, the published summary data (sales by make and sales by region) constrain the possible values of sales figures in the original table.
- *Incorporation of statistical knowledge.* Besides hard constraints, we may have certain statistical knowledge about the data that can help information recovery. Such knowledge may be prior statistical models for the data to be recovered, or noise models of data acquisition, transmission, or perturbation. For example, in IR-RFID, we may know that a book is most likely to be placed on its designated shelf, less likely to be misplaced on a shelf nearby, and least likely to be on some random shelf. From historical data, we may also have some knowledge of the detection error rates of RFID readers. In IR-PPDP, the data publisher may disclose the noise model used in data perturbation. A data analyst may also have some prior knowledge on how sales figures are correlated across makes and regions.
- *Quantification of uncertainty in recovered data.* It is often insufficient for the recovery procedure to offer just one possible reconstruction for the data of interest. Without any measure of uncertainty, a point estimate is difficult to interpret and can be misleading. For example, in IR-RFID, based on the data received, one could return the most likely location of each book; in IR-PPDP, based on the published data, one could return the expected sales for a given make in a given region. However, in both cases, the probability that the returned values agree with actual ones can be very small, because of the sheer number of possibilities. It may be undesirable for applications to make decisions based on low-probability states. To enable confident decisions, it is important to quantify the uncertainty in recovered data.
- *Estimation and representation of distribution of possible reconstructions.* Since there are many possible reconstructions for the data of interest, ideally we would like the recovery procedure to return the distribution of possible reconstructions. The distribution should be represented in a way that captures correlations within data, and supports regular database-style queries over the data. For example, in IR-RFID, because shelves have limited capacity, presence of multiple books on the same shelf are negatively correlated. In IR-PPDP, sales across different makes may be negatively correlated but sales across different regions may be positively correlated. Such correlations would be lost if the the representation used by the recovery procedure is too simple, e.g., one that associates a probability with each possible value for each individual item in the dataset (which would imply that of individual item values are independently distributed).

There is a fundamental trade-off between the generality and efficiency of an information recovery procedure. Previous work on information recovery from the database community tends to lean toward efficiency; as a result, none of existing approaches (to the best of our knowledge) offers all features discussed above. For example, previous approaches based on *entropy maximization (MaxEnt)* incorporate constraints, but it is difficult for them to make use of prior statistical knowledge (e.g., sales follows a Poisson distribution whose rate follows a Gamma prior distribution). For complex scenarios such as IR-RFID [4], we may need to make additional independence assumptions (e.g., book locations are independent) to maintain computational feasibility. As another example, Bayesian analysis has been applied to the IR-PPDP scenario in [5]. The solution is analytic and hence very efficient, but it is specific to one type of prior knowledge (Gaussian perturbation noise). It does not generalize to complex models for which no analytic solutions exist, nor can it incorporate additional constraints (e.g., knowledge of total sales by make and by region).

To overcome the limitations of existing approaches, we take a significantly more general approach to information recovery based on sampling. The high-level idea is simple and intuitive: We generate samples for the reconstructed data, following all known constraints and prior statistical knowledge. Each sample represents one possible reconstruction of the dataset, and the collection of samples represents the distribution of all reconstructions, capturing both uncertainty and correlations.

While the sampling-based approach is general and conceptually clean, it has not been widely applied by the database community to information recovery. Part of the reason may be sampling efficiency. Indeed, with naive sampling, drawing from arbitrary distribution subject to complex constraints can be extremely inefficient as most of the samples may be rejected. In this paper, however, we argue that we can make the sampling-based approach computationally feasible by apply-

ing the technique of *sequential important sampling (SIS)* from statistics [6], [7]. Intuitively, SIS tries to ensure that every sample drawn conforms to known constraints, which dramatically improves sampling efficiency. To correct the sampling bias introduced by SIS, we associate each sample with a *weight*, quantifying the importance of (or the contribution from) the sample in making estimations. These weighted samples can be used to answer queries with uncertainty measures, e.g.: Which shelf is book X most likely on, and with what probability? How likely are books X and Y on the same shelf? What is the probability that the total annual sales of Volvo in North Carolina is between 5,000 and 6,000?

II. Overview of the Approach

We model the original data of interest (which we wish to recover) as a vector of n random variables $\vec{X} = (X_1, X_2, \ldots, X_n)$. Data can be a mix of categorical and numerical data, and can be either discrete or continuous. We call a particular value $\vec{x}$ of $\vec{X}$ a *reconstruction*. Suppose we have some background knowledge about how $\vec{X}$ is distributed over its domain, in term of a *population distribution* with probability density function (pdf) $f(\vec{x})$. In addition, suppose we know a set of constraints $\mathcal{C}$, which must hold on any valid reconstruction of $\vec{X}$. A reconstruction $\vec{x}$ is *valid* with respect to $\mathcal{C}$ if it satisfies all constraints in $\mathcal{C}$, denoted by $\mathcal{C}(\vec{x}) = \texttt{TRUE}$. In information recovery, we are primarily interested in the joint distribution of $\vec{X}$ subject to constraints $\mathcal{C}$. We call the resulting distribution *target distribution* $p(\vec{x})$. Formally, $p(\vec{x}) = f(\vec{x} \mid \mathcal{C}(\vec{x}) = \texttt{TRUE})$.

Instead of returning just one valid reconstruction, our goal is to support reasoning with the target distribution of all valid reconstructions. We return a collection of N weighted samples $(\vec{x}^{(1)}, w^{(1)}), (\vec{x}^{(2)}, w^{(2)}), \ldots, (\vec{x}^{(N)}, w^{(N)})$, where the i-th pair consists of reconstruction $\vec{x}^{(i)}$ and weight $w^{(i)}$. We defer until Section III the discussion on how large N should be.

At a high level, applying this approach to an application involves three steps: 1) *model the application* by formulating the random variables, the population distribution, and constraints; 2) given the population distribution and constraints, *generate the weighted samples*; 3) answer queries and perform analysis *using the weighted samples* generated.

A Naive Method for Generating Samples: Given the population distribution f and constraints $\mathcal{C}$, we need to draw N samples from f satisfying $\mathcal{C}$. Here, we present a simple but practically infeasible method, to help illustrate the purpose of sampling and motivate the need for more efficient techniques.

NaiveSampling$(N, f, \mathcal{C})$

1: **for** $i = 1$ to N **do**
2: **repeat**
3: draw a sample $\vec{x}^{(i)}$ from $f(\vec{x})$;
4: **until** $\mathcal{C}(\vec{x}^{(i)}) = \texttt{TRUE}$;
5: **return** $(\vec{x}^{(1)}, 1), (\vec{x}^{(2)}, 1), \ldots, (\vec{x}^{(N)}, 1)$;

Note that all samples are weighted equally.

There are two major problems with the above sampling algorithm. The first is how to draw samples from $f(\vec{x})$ in the first place. Our goal is to be able to handle arbitrary distributions, but it is difficult to sample directly from complex distributions. The second, and more glaring, problem is its inefficiency, because it must discard all samples that do not satisfy the constraints. This problem deteriorates with more restrictive constraints. As a simple example, consider sampling from a univariate Gaussian distribution $f(x)$ with mean $\mu = 0$ and variance $\sigma^2 = 1$, subject to a constraint $x \geq 5$. The probability of getting a valid sample is extremely small (less than 3×10^{-7}). Hence, much effort is wasted in generating and discarding invalid samples. We address the above two problems using SIS in Section III.

Using Weighted Samples for Queries: After the weighted samples have been generated, we can use them to answer queries. Although the problem of querying incomplete and probabilistic data is not the main focus of this paper, we briefly outline some of the possible queries.

Expectation queries. Formally, suppose we are given a function $h : \vec{x} \to \mathbf{R}$ and we are interested in the expected value of $h(\vec{x})$ when $\vec{x}$ is drawn from the target distribution $p(\vec{x})$, i.e., $\mathbf{E}(h(\vec{x})) = \int h(\vec{x})p(\vec{x})d\vec{x}$. By standard Monte Carlo integration, we can estimate this expectation using N weighted samples $(\vec{x}^{(1)}, w^{(1)}), \ldots, (\vec{x}^{(N)}, w^{(N)})$ as:

$$\hat{h} = \left(\sum_{i=1}^{N} h(\vec{x}^{(i)})w^{(i)}\right) / \left(\sum_{i=1}^{N} w^{(i)}\right).$$

Many interesting questions can be formulated as expectation queries: (1) *Numeric-valued queries.* For any query over the database state represented by $\vec{x}$, if the result is a single numeric value, we can regard the query as a function of $\vec{x}$. For example, in IR-RFID, what is the expected number of books currently on shelf B? In IR-PPDP, what is the expected annual sales of Volvo in North Carolina? (2) *Probability queries.* Such a query asks about the probability that a particular condition holds over the original data of interest. The corresponding function takes $\vec{x}$ to 1 if the condition holds on $\vec{x}$, or 0 otherwise. For example, in IR-RFID, what is the probability that book X is on either shelf C or D? In IR-PPDP, what is the probability that the total annual sales of domestic models in California is greater than that of foreign models?

General database queries. Since each reconstruction $\vec{x}$ represents a possible state of the database, we can evaluate a general query over $\vec{x}$ and produce a answer. The answer produced from a weighted sample is associated with the same weight. The query result over the distribution of possible reconstructions is a distribution of possible answers, naturally represented by the collection of weighted answers.

III. Monte Carlo Sampling

In this section, we describe our approach based on *sequential importance sampling (SIS)* [6], [7], which avoids the problems of naive sampling discussed in the previous section. We start with the statistical foundation, and then address the computational issues in implementing SIS.

A. Statistical Foundations

Importance Sampling: Consider the problem of computing the expectation of query function $h(\vec{x})$ over target distribution, i.e., $\mathbf{E}(h(\vec{x})) = \int h(\vec{x})p(\vec{x})d\vec{x}$. As discussed in Section II, when applying Monte Carlo integration, drawing samples directly from $p(\vec{x})$ may be difficult and inefficient, especially when the distribution is highly constrained. The idea is to pick a different distribution $q(\vec{x})$, called the *trial distribution* to draw samples from instead of $p(\vec{x})$, where $q(\vec{x})$ is easier and more efficient. We are free to choose any trial distribution $q(\vec{x})$ as long as its support[1] is no less than that of the target distribution $p(\vec{x})$, and the ratio $p(\vec{x})/q(\vec{x})$ is bounded. We can ensure proper choice of $q(\vec{x})$ with any distribution having a heavier tail than $p(\vec{x})$. Suppose we draw N samples $\vec{x}^{(1)}, \ldots, \vec{x}^{(N)}$ from the trial distribution. With large enough N, the following converges to $\mathbf{E}(h(\vec{x}))$ by the Strong Law of Large Numbers:

$$\bar{h}_N = \frac{\sum_{i=1}^{N} h(\vec{x}^{(i)})\frac{p(\vec{x}^{(i)})}{q(\vec{x}^{(i)})}}{\sum_{i=1}^{N}\frac{p(\vec{x}^{(i)})}{q(\vec{x}^{(i)})}} = \frac{\sum_{i=1}^{N} h(\vec{x}^{(i)})w^{(i)}}{\sum_{i=1}^{N} w^{(i)}}. \quad (1)$$

In the above, $w^{(i)} = p(\vec{x}^{(i)})/q(\vec{x}^{(i)})$ is the *weight* of sample $\vec{x}^{(i)}$, which corrects the bias introduced by sampling from the trial distribution, and quantifies the contribution of $\vec{x}^{(i)}$ in estimation. Note that the same collection of samples and weights can be used to evaluate different query functions. Furthermore, we only need to compute weights up to some unknown constant (i.e., all weights can be scaled by the same constant); in (1) above, the unknown constant in numerator and denominator will cancel out each other.

How many samples do we need? The answer in general depends on the query function $h(\vec{x})$ we are interested in computing. Practically, with N weighted samples, we can estimate the standard error in estimation by $\sqrt{\sum_{i=1}^{N}(h(\vec{x}^{(i)})w^{(i)})^2 - N(\bar{h}_N)^2}/(\sum_{i=1}^{N} w^{(i)})$. If error is too high, we acquire additional samples.

To measure the effectiveness of importance sampling, we can compare the number of samples needed from the trial distribution (when using importance sampling) with the number of samples needed from the target distribution (when using basic sampling), in order to reach the same estimation accuracy. This comparison in general also depends on the query function. A practical and "function-free" alternative is to use *effective sample size (ESS)*, a measure proposed in [8]. Specifically, ESS $= \frac{N}{1+cv^2}$, where the *coefficient of variation*, cv^2, can be approximated from the sample weights as follows:

$$cv^2 = \frac{\mathbf{var}_q(p(\vec{x})/q(\vec{x}))}{\mathbf{E}_q^2(p(\vec{x})/q(\vec{x}))} \approx \frac{\frac{1}{N-1}\sum_{i=1}^{N}(w^i - \sum_{i=1}^{N} w^{(i)}/N)^2}{(\sum_{i=1}^{N} w^{(i)}/N)^2}.$$

Intuitively, cv^2 measures how far the trial distribution is to the target distribution. Small cv^2 implies that trial and target distributions are similar, and therefore importance sampling will not need many more samples. For example, if $cv^2 = 2$, then $3M$ samples from the trial distribution should provide comparable accuracy as M samples from the target distribution.

[1]The support of a distribution $f(x)$ is the smallest closed set X such that $f(x) \neq 0, \forall x \in X$.

Sequential Importance Sampling (SIS): Recall that we need to reconstruct a vector $\vec{x}$ rather than a single value. Since it generally difficult to sample all elements simultaneously from a multivariate distribution, we sample the components of $\vec{X}$ *sequentially* in some order $X_1, X_2, \ldots, X_n$. The trial distribution can be written as a chain product:

$$q(\vec{x}) = q_1(x_1)q_2(x_2|x_1)\ldots q_n(x_n|x_1, x_2, \ldots, x_{n-1}). \quad (2)$$

We first sample X_1 from $q(x_1)$. Subsequently, we sample X_k conditioned on the previously sampled values $x_1, \ldots, x_{k-1}$, i.e., from $q_k(x_k|x_1, \ldots, x_{k-1})$. The trial distribution is chosen such that the conditional distributions are easy to sample from. The weight of a complete sample $\vec{x} = (x_1, \ldots, x_n)$ is computed as:

$$w(\vec{x}) = \frac{p(\vec{x})}{q_1(x_1)\prod_{k=2}^{n} q_k(x_k|x_1, x_2, \ldots, x_{k-1})}. \quad (3)$$

The weighted samples obtained by SIS are used for queries in the same way as those obtained by importance sampling.

B. Computational Issues in Applying SIS

In this section, we discuss various computational issues that arise in implementing SIS and how we address them. First, we start with the generic SIS algorithm from [9]. Recall that the n-dimensional target distribution p is given by the population distribution f subject to a set of constraints $\mathcal{C}$. For simplicity, we assume that all values are integers; it is straightforward to extend the algorithm to other discrete domains (by mapping them to integers) and to reals.

SIS$(N, f, \mathcal{C})$

```
1:  for i = 1 to N do
2:    repeat
3:      valid ← true, w ← 1, C_0 ← C;
4:      for k = 1 to n do
5:        I_k ← BOUND(C_{k-1}, k);
6:        if LENGTH(I_k) = 0 then
7:          valid ← false;
8:          break;
9:        x_k ← UNIFORM(I_k);
10:       w ← w × LENGTH(I_k);
11:       if k < n then
12:         C_k ← UPDATECONSTRAINTS(C_{k-1}, k, x_k);
13:   until valid
14:   x^(i) ← (x_1, ..., x_n), w^(i) ← w · f(x);
15: return (x^(1), w^(1)), (x^(2), w^(2)), ..., (x^(N), w^(N));
```

To obtain each sample, we use function BOUND (Line 5; more on it later) to compute from the given set of constraints a bound on the possible values for x_1, the first component of the sample. We then draw a value for x_1 uniformly from this bound (Line 9). Though the algorithm can be easily

modified to use other trial distributions, we have found that a chain of uniform conditional distributions works well for our applications. After drawing the k-th component x_k, we use function UPDATECONSTRAINTS to update the constraints (Line 12); specifically, it substitutes every occurrence of x_k in the constraints with the actual value drawn. The bound for the next component will be calculated using the updated constraints, and the process repeats.

It is still possible to obtain a bad sample, e.g., when a bound does not contain any integer. This case is detected by checking (Line 6) whether the bound I_k derived for x_k has zero length (calculated as the number of integers within I_k). In that case, we discard the partial sample and start over. By design, SIS generates dramatically fewer bad samples than naive sampling, as we will experimentally verify.

Calculation of the sample weight $w^{(i)}$ is incremental (Lines 10 and 14), and reflects the use of a chain of uniform conditional distributions as the trial distribution. Also, note that this $w^{(i)}$ is not exactly $p(\vec{x}^i)/q(\vec{x}^i)$, but instead scaled by $\int_{C(\vec{x})=\texttt{TRUE}} f(\vec{x})d\vec{x}$. We can avoid calculating this integral because, as discussed in Section III-A, all sample weights are scaled by the same value.

For complex (e.g., non-linear) constraints, BOUND may require expensive constraint solving. In general, the bound may also turn out to be a set of intervals instead of a single interval. However, we observe that many constraints that arise in practice, including our motivating applications IR-RFID and IR-PPDP, are linear. This observation prompts us to focus on linear constraints, which we discuss below.

B.1) Optimizations for Linear Constraints

A set of m linear equality constraints over a sample $\vec{x}$ with n components can be encoded by a pair $(\mathbf{A}, \vec{b})$, where $\mathbf{A}$ is an $m \times n$ *constraint matrix*, and $\vec{b}$ is a *requirement vector* of length m. A valid sample $\vec{x}$ should satisfy $\mathbf{A}\vec{x} = \vec{b}$. Inequality constraints can be handled using the well-known trick of extra *slack variables*. We further assume that all variables are implicitly non-negative; negative variables can be handled through variable transformation. Details on how to handle slack and negative variables can be found in [10].

UPDATECONSTRAINTS, which updates the constraints after a new component x_k has been sampled, can be implemented using basic matrix operations. Specifically, let $(\mathbf{A}_k, \vec{b}_k)$ denote the set of updated constraints after $x_1, \ldots, x_k$ has been sampled. Then:

- $\mathbf{A}_k = \mathbf{A}_{k-1}[, -1]$; i.e., remove $\mathbf{A}_{k-1}$'s first column.
- $\vec{b}_k = \vec{b}_{k-1} - x_k \mathbf{A}_{k-1}[, 1]$, where $\mathbf{A}_{k-1}[, 1]$ denotes the first column of $\mathbf{A}_{k-1}$.

We consider three approaches for BOUND, which computes the bound on x_k from $(\mathbf{A}_{k-1}, \vec{b}_{k-1})$.

- ***LP***: This approach uses linear programming (LP) to minimize and maximize x_k subject to the given constraints. For solving LP, there exist polynomial-time algorithms (e.g., Khachiyan's ellipsoid method and Karmarkar's interior-point method) as well as the simplex algorithm, which is exponential in the worst case but works well in practice.
- ***SHUTTLE***: This algorithm, proposed by [11], finds the bound on each variable by iterating through variable dependencies. For n variables and m constraints, SHUTTLE takes $O(kmn^3)$ time, where k is the number of iterations it uses. However, the bounds computed by SHUTTLE may not be tight, which may lead to more bad samples.
- ***DIRECT***: If the data to be reconstructed consists of non-negative entries from a table, and all constraints are row sums and column sums in this table, then we can apply a much more efficient algorithm from [12] to directly calculate the sampling interval. Suppose the sum of all entries in each row i is r_i, and the sum of all entries in each column j is c_j. Let T be the sum of all entries, which can be computed as either $\sum_i r_i$ or $\sum_j c_j$. Let $x_{i,j}$ denote the non-negative table entry we need to recover at row i, column j. It is not difficult to see that $x_{i,j} \in [\max\{0, r_i + c_j - T\}, \min\{r_i, c_j\}]$. Once we have sampled $x_{i,j}$, we decrement r_i, c_j, and T each by the value of $x_{i,j}$. These quantities are then used to compute bounds on remaining entries to be sampled. This approach computes a tight bound in $O(1)$ time.

Assuming simple trial distributions (such as uniform), the cost of Algorithm SIS is dominated by computing bounds. In Section VI, we show that the optimizations above allow sampling to scale to very large problems, especially if efficient bound calculation algorithms such as DIRECT are applicable (which is the case for IR-RFID, for example).

IV. APPLICATION IN IR-RFID

We now show how to apply our approach to the problem of recovering information from noisy RFID data, introduced in Section I. We first discuss our approach, focusing on how to model the application. We then describe an alternative approach based on MaxEnt used by [4]. Later, in Section VI-A, we experimentally compare the two approaches in terms of the quality of their reconstruction.

A. Our Approach

Recall that in IR-RFID, we use RFID detectors on library shelves to monitor book locations. In each timestep, every shelf reports all books detected on that shelf to a base station. Suppose there are n_s shelves and n_b books. For the current timestep, let indicator variable $X_{i,j}$ denote whether book i is indeed placed on shelf j ($1 \le i \le n_b$, $1 \le j \le n_s$). Specifically, $X_{i,j} = 1$ if book i is on shelf j; otherwise, $X_{i,j} = 0$. To account for the possibilities that a book is checked out, we also include a special "shelf" for holding all checked-out books. The state vector $\vec{X} = (X_{1,1}, \ldots, X_{n_b,n_s})$ represents the "truth," which we do not have direct access to but would like to recover from observations, which include the RFID readings received at the base station as well as information about which books are currently checked out.

Population Distribution: We now seek the population distribution for $\vec{X}$. In order to incorporate the information from observations, we model the observation results using another set of random variables $Y_{i,j}$, where $Y_{i,j} = 1$ if book i is detected by shelf j's sensor; otherwise, $Y_{i,j} = 0$. Suppose we have an error model for the RFID detectors, we can incorporate this knowledge in a form of a conditional distribution with pdf $p(\vec{y}|\vec{x})$, which returns the probability of observing $\vec{Y} = \vec{y}$ given the true state $\vec{X} = \vec{x}$.

As a simple example, suppose that $Y_{i,j}$ only depends on $X_{i,j}$, and the detector for shelf j has false-negative rate α_j (the probability that a book on the shelf is undetected) and false-positive rate β_j (the probability that a book not on the shelf is falsely detected). Then $p(\vec{y}|\vec{x}) = \prod_{i,j} \big(x_{i,j}((1-\alpha_j)y_{i,j} + \alpha_j(1-y_{i,j})) + (1-x_{i,j})(\beta_j y_{i,j} + (1-\beta_j)(1-y_{i,j}))\big)$. For the special "checked-out" shelf, we can set the false-negative and false-positive rates to 0, assuming that the information about checked-out books (presumably from a circulation database) is reliable. Note that the assumptions above is only for simplicity of presentation. In general, detections errors may not be independent, and we would have a more sophisticated joint distribution $p(\vec{y}|\vec{x})$; our approach would still be applicable.

Additionally, we can capture any prior knowledge of where the books are currently located with the distribution $p(\vec{x})$. For example, this distribution can be dependent on the book locations in the previous timestep—books tend to stay where they were. This distribution can also incorporate the knowledge of how books are supposed to be shelved according to catalogs—books are more likely to be on their designated shelves. In the worst case when no information is available, we can simply assume that a book appears on each shelf with equal probability. Our approach can readily handle setups with varying complexities.

Now, given an observation vector $\vec{y}$ at the current time, we can define the population distribution for the true current state as $p(\vec{x}|\vec{y})$ (conditioned on $\vec{y}$). By Bayes' law, $p(\vec{x}|\vec{y}) \propto p(\vec{y}|\vec{x})p(\vec{x})$. Note that we simply use $p(\vec{y}|\vec{x})p(\vec{x})$ instead of the actual $p(\vec{x}|\vec{y})$, which would require computing $p(\vec{y})$. The reason is that all sample weights are scaled by the same $p(\vec{y})$, which will be canceled out in estimation (Section III).

Constraints: The following constraints naturally arise:

- (C1) A book cannot be on two shelves at the same time, or be on a shelf and checked out; i.e., $\sum_{j=1}^{n_s} x_{i,j} \le 1$.
- (C2) The number of books on a shelf j cannot exceed its maximum capacity k_j; i.e., $\sum_{i=1}^{n_b} x_{i,j} \le k_j$.

Others can be incorporated but are omitted for brevity.

Sampling: Algorithm SIS can be used to draw samples efficiently from distribution $p(\vec{x}|\vec{y})$ subject to the constraints above. Note that all variables are non-negative (0 or 1) and form an $n_b \times n_s$ table, and the constraints above are row and column sums (with the introduction of slack variables). Therefore, we can use the efficient DIRECT procedure (Section III-B.1) for computing bounds.

B. Alternative Approach Based on MaxEnt

The approach of [4] is based on MaxEnt. On the high level, MaxEnt is method for finding a "least biased" probability distribution that satisfies a set of constraints. Given a discrete probability space, MaxEnt assigns a probability $p(\vec{v})$ to each event $\vec{v}$ such that:

- $p(\vec{v})$ is a probability distribution; i.e., $\sum_{\vec{v}} p(\vec{v}) = 1$.
- A given set of constraints $\mathcal{C}$ over the probability assignments, $\mathcal{C}(\{p(\vec{v})\})$, is satisfied.
- The entropy of the distribution, $-\sum_{\vec{v}} p(\vec{v}) \log p(\vec{v})$, is maximized.

There exist efficient algorithms for this problem when the constraints are simple (e.g., linear), and they have been applied by the database community (e.g., [13], [3]).

Direct application of MaxEnt to IR-RFID is computationally infeasible, however, because the space of possible states is huge—with n_b books and n_s shelves, there are up to $(n_s+1)^{n_b}$ probabilities for MaxEnt to assign. Instead, the approach of [4] uses MaxEnt to assign each $p_{i,j}$, the (marginal) probability that book i is at location j (including special "check-out" and "unshelved" locations). Hence, the total number of probabilities to assign is reduced to $O(n_b \times n_s)$. Each book i must be at some location, so $\sum_j p_{i,j} = 1$. Constraint (C1) is implicit; constraint (C2) involving maximum shelf capability k_j can be roughly encoded as $\sum_{i=1}^{n_b} p_{i,j} \le k_j$. Note that this encoded constraint is actually weaker; it literally means that the *expected* book count on shelf j is no more than k_j, while the original (C2) dictates that the count is absolutely no more than k_j.

Information about checked-out books and received RFID readings are incorporated as additional constraints:

1) If book i is known to be checked out, then $p_{i,c} = 1$, where c denotes the special "checked-out" location; otherwise, $p_{i,c} = 0$.
2) If book i is detected on exactly one shelf j and we know it is not checked out, then $p_{i,j} = 1$; i.e., we trust the reading completely.
3) If book i is simultaneously detected on multiple shelves, these readings are effectively ignored.

While this approach provides a pragmatic way of incorporating prior knowledge, due to limitations of MaxEnt, available knowledge is not fully captured. For example, the second case above ignores the possibility of false positives; the third discards too much information in resolving conflicts.

In conclusion, the MaxEnt-based approach of [4] provides an efficient and practical solution to information recovery, but it is not without limitations. First, being based on MaxEnt, it is unable to capture all knowledge that our sampling-based approach is able to incorporate in Section IV-A. Second, efficiency comes at a cost. The representation of a joint distribution by marginal probabilities causes much of the dependencies to be lost. For example, the reconstructed distribution may indicate that books X and Y each have probability 0.5 to be on shelf A (with one remaining slot), but fail to capture the fact that they cannot be on A at the same time.

V. Application in IR-PPDP

A. Our Approach

Consider again the scenario of publishing a two-dimensional auto sales table T with $n = n_m \times n_r$ entries where the entry at (i, j) stores the annual sales of make i in region j. We model the n entries of table T as a random vector $\vec{X}$ of size n, whose value we need to reconstruct. By abuse of notation, let x_i denote the i-th component of $\vec{x}$, and $x_{i,j}$ denote the component of $\vec{x}$ corresponding to the entry of T at (i, j).

Recall that in IR-PPDP, the data publisher adds a random noise (with n components) to the original data and publishes the resulting data, which we denote by a vector $\vec{z}$. Since we do not know the actual noise added, we model it as random vector $\vec{Y}$ of size n.

Our goal is reconstruct $\vec{X}$, given the perturbed data $\vec{z}$ as well as any additional information obtained from other sources, e.g., the total sales of GM across all regions, or the fact that in North Carolina, the combined sales of all domestic makes is higher than that of all foreign makes.

Population Distribution: We can use the population distribution $f_{\vec{X},\vec{Y}}(\vec{x}, \vec{y})$ to encode any prior knowledge about the original table and about the perturbation noise. If noise was produced independently from the original data, then $f_{\vec{X},\vec{Y}}(\vec{x}, \vec{y}) = f_{\vec{X}}(\vec{x}) f_{\vec{Y}}(\vec{y})$, where $f_{\vec{X}}(\vec{x})$ captures our knowledge about the original data to be reconstructed, while $f_{\vec{Y}}(\vec{y})$ captures our knowledge of perturbation process. Again, our approach can handle prior knowledge with varying degree of uncertainty and complexity.

Constraints: There are two types of constraints. The first type captures the relationship among the original data $\vec{x}$ (hidden), the perturbation noise $\vec{y}$ (hidden), and perturbed data $\vec{z}$ (published). Specifically, $x_i + y_i = z_i$ for every i.

Constraints of the second type, defined on $\vec{x}$, come from other public information. For example, we may know the total sales of make i across all regions to be c_i; the constraint would be $\sum_j x_{i,j} = c_i$. For another example, we may know that in region j, the combined sales of all domestic makes (set D) is higher than that of all foreign makes (set F); the constraint would be $\sum_{i \in D} x_{i,j} > \sum_{i \in F} x_{i,j}$.

Sampling: Algorithm SIS can be used to draw samples efficiently from $f_{\vec{X},\vec{Y}}(\vec{x}, \vec{y})$ subject to both types of constraints above. Note that all constraints are linear (though they are not all row and column sums), so we can use LP or SHUTTLE (Section III-B.1) for computing bounds.

B. Alternative Analytic Solution

A Bayesian approach for recovering original data from perturbed data is proposed in [5]. This approach models the original data using a Gaussian prior distribution $f(\vec{x})$, whose mean and variance are estimated from the perturbed data $\vec{z}$ and knowledge of the perturbation procedure. The perturbation procedure is modeled using a conditional distribution $f(\vec{z}|\vec{x})$, again Gaussian. The information recovery procedure may produce the posterior distribution $f(\vec{x}|\vec{z})$ as the distribution of possible reconstructions. In [5], an efficient analytic solution is used for finding the maximum a posteriori (MAP) estimate of $\vec{x}$. Such a solution is possible because both $f(\vec{x})$ and $f(\vec{z}|\vec{x})$ are Gaussian.

Although analytic solutions are efficient, existence of such depends on the forms of distributions involved in analysis. They are difficult to generalize to other distributions, more complex combinations of prior and likelihood, and additional constraints. In particular, the data publisher can easily render the approach inapplicable by choosing a noise distribution that is difficult to work with analytically. Additional knowledge in the form of constraints from Section V-A cannot be incorporated either.

VI. Experiments

The first two groups of experiments we present functionally compare our sampling-based approach with previous approaches, for applications IR-RFID and IR-PPDP. Most of these experiments have small scales, because we want to make the results intuitive and easy to compare. The third group of experiments are larger in scale and are designed to demonstrate the efficiency and scalability of our approach. All experiments are conducted a Dell Dimension 8300s with 3.0GHz Intel Pentium CPU and 1GB memory.

A. IR-RFID Functional Experiments

We simulate a system that collects RFID readings for tracking locations of books in a library, as discussed in Section IV-A. To make the results easier to interpret, we set the false positive rate to 0; i.e., if a book is detected on a shelf then it must be on that shelf. False negatives are still possible; i.e., if a book is on a shelf it still may be undetected. Even with this simplified problem setup, we can demonstrate clear advantage of our approach over the approach of [4] (henceforth abbreviated as MaxEnt).

To make the results easy to understand, we use a small setup where we want to determine the locations of 12 undetected books among 4 shelves A, B, C, and D with remaining capacity. The detectors on shelves B and C are very reliable, with 0.01 failure probability (false negative rate). For the detectors on shelves A and D, we vary their failure probabilities from 0.01 (very reliable) to 0.32 (unreliable). For comparison, we plot the location distributions for two books as reconstructed by SIS from 5000 samples and MaxEnt for each configuration.

Uniform Prior, Uneven Shelf Capacity: First we show SIS can incorporate both statistic knowledge (detection failure probabilities) and hard constraints (remaining shelf capacity). We assume uniform prior distribution of book locations. The remaining capacities for shelves A, B, C, D are $1, 2, 4, 5$ respectively (after accounting for books detected on shelves). Figure 2 shows the reconstructed location distributions for two books as we vary the failure probabilities of A and D from 0.01 to 0.32 in smaller figures.

Across the small figures, we see that MaxEnt produces the same distribution for both books (shown by the same bars) across all failure probabilities; i.e., it ignores prior knowledge

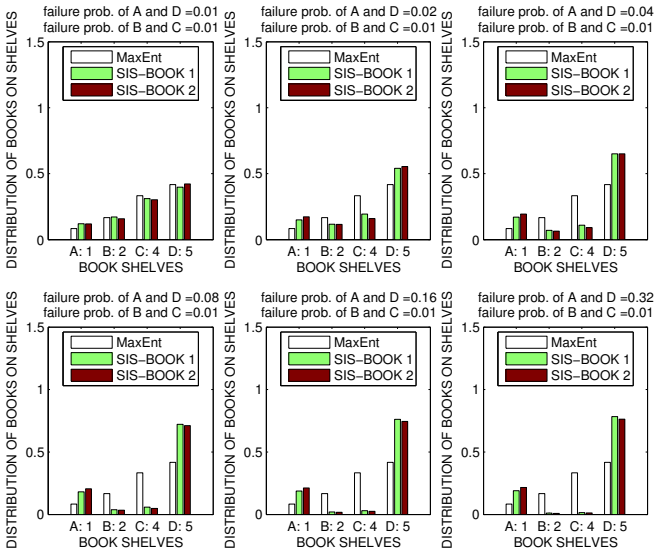

Fig. 2. Uniform prior, uneven shelf capacity.

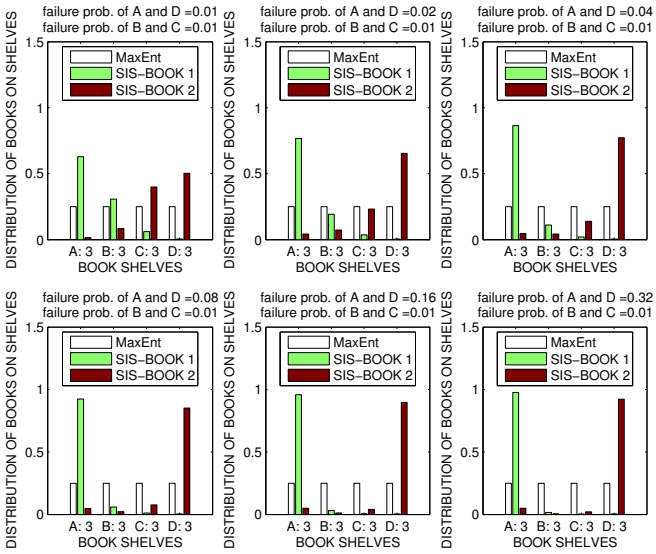

Fig. 3. Gaussian prior, even shelf capacity.

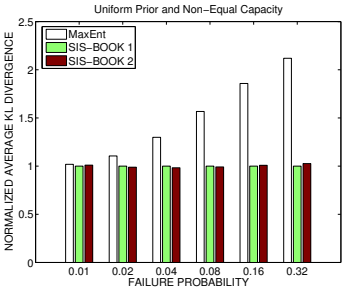

Fig. 4. Uniform, uneven capacity.

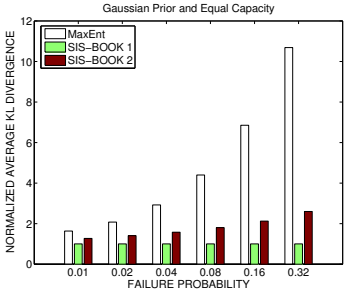

Fig. 5. Gaussian, even capacity.

of failure probabilities. In comparison, our SIS approach incorporates both constraints and knowledge of failure probabilities. When the failure probabilities of shelves A and D increase, SIS is able to differentiate them from B and C, which remain very reliable. Higher failure probabilities of A and D cause SIS to allocate higher probabilities for books to be on A and D, because it would be more likely for A and D to miss the detection than B and C. Between A and D, we still see probabilities being allocated proportionally according to their capacities, indicating that SIS also considers capacity constraints.

Gaussian Prior, Even Capacity: To demonstrate how SIS can take advantage of more informative prior knowledge, suppose that each book is most likely to be placed on its designated shelf, and that the probability of being misplaced on another shelf decreases with the distance from the designated shelf; we use a discretized Gaussian to encode this knowledge. Suppose we also know that A is the designated shelf of book 1, and D is the designated shelf of book 2. Furthermore, book 2 tends to be misplaced more often than book 1 (e.g., because book 2 is more popular), which is captured by a higher variance in the prior location distribution of book 2. To make the results easier to interpret, we assume all four shelves have the same remaining capacity (of 3 books). We again increase the failure probabilities of A and D from 0.01 to 0.32 and show the results in Figure 3.

As MaxEnt incorporates neither the prior knowledge of location distribution nor the failure probabilities, it assigns probabilities uniformly across shelves and never distinguishes the two books (again shown by the same bars). SIS, however, considers both forms of prior knowledge. Intuitively, the reconstructed distribution shows how the probability drops as distance to the designated shelf increases (we assume that shelves A, B, C, D are located on a line in order). As detection on shelves A and D (which happen to the designated shelves for the two books of interest) becomes more unreliable, the belief that the books on these shelves is strengthened, as it is less likely for the books to be undetected on other shelves.

Accuracy of Reconstruction: Ultimately, we are interested in how close the reconstructions provided by SIS and MaxEnt are to the true state of book locations.To measure the error in a reconstructed distribution relative to the true state, we use the standard *Kullback-Leibler distance (KL-distance)*. We regard the true state as a probability distribution in which the true state occurs with probability 1, and measure the KL distances from the distributions reconstructed by SIS and MaxEnt to this distribution. For each of the above two experiment configurations, we test SIS and MaxEnt with 5000 different true states, and in each test we measure the KL distances from the location distributions of books 1 and 2 reconstructed from 5000 samples to the true state. We report the average KL distances over 5000 tests in Figures 4 and 5. The distances shown are always normalized according to the KL distance between the true state and book 1's distribution as reconstructed by SIS. Again, since MaxEnt does not differentiate the two books, the results for the two books are the same for MaxEnt across all configuration and hence shown by the same bars. SIS consistently provides better accuracy than MaxEnt for both books.

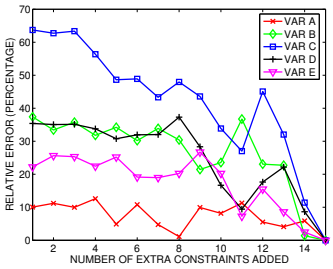

Fig. 6. Incorporating constraints.

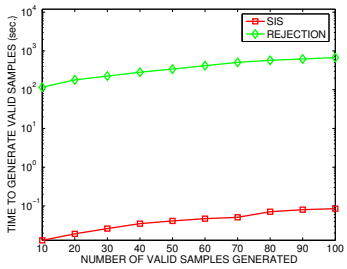

Fig. 7. Sampling efficiency.

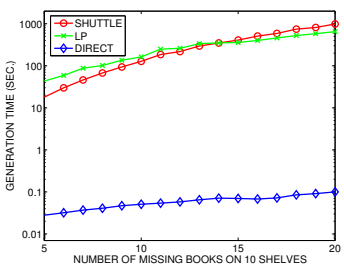

Fig. 8. Bound calculation.

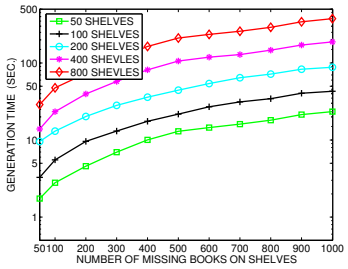

Fig. 9. Larger problem sizes.

B. IR-PPDP Functional Experiments

For information recovery from perturbed data and constraints, we compare our SIS-based approach with the Bayesian approach of [5] based on an analytic solution (henceforth abbreviated as analytic Bayesian). Because of space constraints, we present only one set of experiments showing how SIS can effectively exploit constraints; additional experiments can be found in [10]. We consider information recovery for a 5×5 table with 25 sensitive entries. We introduce extra constraints in addition to the 25 basic constraints that we use to capture the relationship among original data, perturbation noise, and published data, as discussed in Section V-A. There are up to 5 row-sum and 5 column-sum constraints; we also introduce up to 5 random constraints on the sum of random entries in the table. We introduce these extra constraints in an incremental fashion, and for each set of extra constraints, we use SIS to generate 1000 weighted samples. For each table entry x, we compute the relative error between the expectation of x computed by SIS and the actual value of x before perturbation. We pick five random entries and show how their relative errors change as extra constraints are added in Figure 6. Generally, errors tend to reduce with more constraints, because they limit the possible choices of values. On the other hand, analytic Bayesian cannot incorporate extra constraints, so its estimation errors remain high at the point where there are no extra constraints available.

C. Performance Experiments

SIS vs. Naive Sampling: This set of experiments aims at demonstrating the dramatic efficiency improvement of SIS over naive sampling (Section II). The application scenario is IR-PPDP; we obtain the light vehicle sale figures in US by companies from January through September in years 2005 and 2006. We consider a very simple 4×2 table with four extra constraints; see [10] for detailed setup. To compare efficiency we measure the time it takes to generate the same number of valid samples for naive sampling and for SIS. Figure 7 compares the time to generate 10 to 100 valid samples. Even with such a small problem, naive sampling is extremely slow, because most samples it produces are inconsistent with constraints and therefore discarded. In particular, it takes naive sampling around 670 seconds to obtain 100 valid samples out of around $500,000$ samples generated, while it takes SIS less than 0.085 second to generate the same number of samples (a speedup-factor of 8000).

Scalability: To demonstrate the scalability of SIS and the computational issues affecting its scalability, we first compare the three bound calculation algorithms: LP, SHUTTLE, and DIRECT. For LP, we used the *lp_solve* package developed by Michel Berkelaar. The application scenario is IR-RFID. We first consider a setup that requires recovering the locations of undetected books among 10 shelves with remaining capacity. In Figure 8, we increase the number of undetected books from 5 to 20 and show the time for SIS to draw 5000 valid samples (sufficient for good reconstruction), with different bound calculation algorithms. As shown in Figure 8, DIRECT outperforms LP and SHUTTLE by orders of magnitude. In particular, for 20 undetected books, DIRECT takes less than 0.1 second to generate all 5000 valid samples, a speedup-factor of 10000 over LP and SHUTTLE. SHUTTLE works better than LP when the problem scale is small. However, recall from Section III-B.1 that the bounds computed by SHUTTLE may not be tight, which may lead to more bad samples. Indeed, when the number of undetected books increases, SHUTTLE begins to lag behind LP because of this problem.

Next, we test the scalability of SIS with DIRECT on much larger problems. We simulate 1000 undetected books, and increase the number of available shelves from 50 to 800. There are as many as $800,000$ values to be reconstructed. We measure the time to generate $10,000$ valid samples, sufficient for the largest problem in this case.[2] In Figure 9, we show the running time as we increase the number of available shelves from 50 to 800. We see that we can generate $10,000$ samples (each of which involves sampling $800,000$ variables) in just 400 seconds. Problems of this scale would be challenging for MaxEnt and analytic Bayesian, not to mention that they cannot incorporate prior statistical knowledge and constraints simultaneously as SIS does.

VII. Related Work

Traditional database sampling deals with the problem of sampling from a large dataset, while our approach is about drawing samples (that conceptually represent database states) from distributions. In the statistics community, sequential Monte Carlo methods have been proposed as a general data

[2] To verify that $10,000$ samples are indeed sufficient for the largest problem with 1000 undetected books and 800 shelves, we generate $1,000,000$ samples and use them to reconstruct the location distribution for randomly chosen books. We compare this distribution with one obtained using $10,000$ samples, and found their KL distance to be less than 0.001.

augmentation procedure to solve Bayesian inference problems [8]. In this paper, we apply similar ideas to problems of interest to the database community.

In the database community, research on information recovery can be traced back to early 1980s. Approaches in [14], [15] do not offer statistical confidence of its answers to users. In [16], Fatoutsos et al. studied the problem of recovering a original table given partial sums; our approach is more general and can incorporate general constraints as well as prior statistical knowledge. More importantly, [16] and many previous approaches reconstruct only *one* possible instance that maximizes some objective function, e.g., entropy (MaxEnt). We argue that for many applications, what is really needed is the distribution of possible reconstructions, which supports much richer queries (Section II).

The two application areas tackled by this paper, sensor/RFID networks [17], [1] and privacy-preserving data publishing [18], [2], [19], [20], have received much attention from the database community recently. Previous research on recovering information from noisy and incomplete data in these settings [4], [5] recognizes the importance of using prior knowledge, which can be hard constraints or soft statistical knowledge. However, as far as we know, no previous work was able to incorporate both types of knowledge simultaneously in a general and principled way; our paper fills this gap.

The importance of handling data uncertainty has also recently prompted series of work on querying probabilistic data [21], [22], [23]. These approaches focus on developing tractable query processing algorithms on certain efficient representations of probabilistic data. On the other hand, our general sampling-based approach can work on distributions with unknown or complex structures subject to constraints. These approaches complement ours in the spectrum of generality/efficiency trade-off.

VIII. Conclusion

In this paper, we propose a sampling-based approach to recovering information from noisy and incomplete data. Our approach is able to simultaneously exploit known constraints and prior statistical knowledge about the data. While computationally more efficient alternatives may exist for special cases, our sampling-based approach is general in that it does not assume specific forms of distributions and constraints. Our approach provides the distribution of possible reconstructions, represented as a collection of weighted samples. Sampling efficiency is achieved using SIS, which works for highly complex distributions and dramatically outperforms naive sampling methods when data is constrained. We address the important computational issues in implementing SIS, and show that it scales well with efficient bound calculation algorithms. We illustrate the power, generality, and efficiency of our approach using two application scenarios: cleansing RFID data, and recovering information from published data that has been summarized or randomized for privacy. Finally, we also note that sampling is embarrassingly parallel, which makes it easy to scale up with additional computational resources.

References

[1] S. R. Jeffery, M. Garofalakis, and M. Franklin, "Adaptive cleaning for rfid data streams," in *Proc. of the 2006 Intl. Conf. on Very Large Data Bases*, Seoul, Korea, June 2006.

[2] R. Agrawal and R. Srikant, "Privacy-preserving data mining," in *Proc. of the 2000 ACM SIGMOD Intl. Conf. on Management of Data*, Dallas, TX, USA, May 2000.

[3] U. Srivastava, P. Hass, V. Markl, N. Megiddo, M. Kutsch, and T. Tran, "Isomer: Consistent histogram construction using query feedback," in *Proc. of the 2006 Intl. Conf. on Data Engineering*, Atlanta, Geogia, USA, Apr. 2006.

[4] N. Khoussainova, M. Balazinska, and D. Suciu, "Towards correcting input data errors probabilistically using integrity constraints," in *Proc. of the 2006 ACM Workshop on Data Engineering for Wireless and Mobile Access*, Chicago, Illinois, USA, June 2006.

[5] Z. Huang, W. Du, and B. Chen, "Deriving private information from randomized data," in *Proc. of the 2005 ACM SIGMOD Intl. Conf. on Management of Data*, Baltimore, Maryland, USA, June 2005.

[6] J. S. Liu and R. Chen, "Sequential Monte-Carlo methods for dynamic systems," *J. American Statistical Association*, vol. 93, pp. 1032–1044, 1998.

[7] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. Springer, 2001.

[8] A. Kong, J. S. Liu, and W. H. Wong, "Sequential imputation method and bayesian missing data problems," *Journal of the American Statistical Association*, vol. 89, no. 278-288, 1994.

[9] Y. Chen, I. H. Dinwoodie, and S. Sullivant, "Sequential importance sampling for multiway tables," *The Annals of Statistics*, vol. 34, 2005.

[10] J. Xie, J. Yang, Y. Chen, H. Wang, and P. S. Yu, "A sampling-based approach to information recovery," Duke University, Tech. Rep., Mar. 2007, http://www.cs.duke.edu/dbgroup/papers/2007-xycwy-sampling.pdf.

[11] L. Buzzigoli and A. Giusti, "An algorithm to calculate the lower and upper bounds of the elements of an array given its marginals," in *Proc. of the Conference on Statistical Data Protection*, Luxembourg, 1999, pp. 131–147.

[12] Y. Chen, P. Diaconis, S. P. Holmes, and J. S. Liu, "Sequential monte carlo methods for statistical analysis of tables," *Journal of the American Statistical Association*, vol. 100, no. 109-120, 2005.

[13] V. Markl, N. Megiddo, M. Kutsch, T. M. Tran, P. J. Haas, and U. Srivastava, "Answering queries from statistics and probabilistic views," in *Proc. of the 2005 Intl. Conf. on Very Large Data Bases*, Trondheim, Norway, Aug. 2005.

[14] H. Sato, "Handling summary information in a database: Derivability," in *Proc. of the1981ACM SIGMOD Intl. Conf. on Management of Data*, Ann Arbor, Michigan, USA, Apr. 1981.

[15] F. M. Malvestuto, "A universal-scheme approach to statistical databases containing homogeneous summary tables," *ACM Transaction on Database Systems*, vol. 18, no. 678-708, 1993.

[16] C. Faloutsos, H. V. Jadadish, and N. D. Sidiropoulos, "Recovering information from summary data," in *Proc. of the 1997 Intl. Conf. on Very Large Data Bases*, Athens, Greece, Sept. 1997.

[17] A. Deshpande, C. Gustrin, S. R. Madden, J. M. Hellerstein, and W. Hong, "Model-based approximate querying in sensor networks," *VLDB Journal*, 2005.

[18] P. Chu, "Cell suppression methodology: The importance of suppressing marginal totals," *IEEE Trans. on Knowledge and Data Engineering*, vol. 9, no. 4, pp. 513–523, 1997.

[19] D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *Proc. of the 2001 ACM Symp. on Principles of Database Systems*, Santa Babara, CA, USA, June 2001.

[20] D. Kifer and J. Gehrke, "Injecting utility into anonymized datasets," in *Proc. of the 2006 ACM SIGMOD Intl. Conf. on Management of Data*, Chicago, Illinois, USA, June 2006.

[21] N. Dalvi and D. Suciu, "Efficient query evaluation on probabilistic databases," in *Proc. of the 2004 Intl. Conf. on Very Large Data Bases*, Toronto, Canada, Aug. 2004.

[22] ——, "Answering queries from statistics and probabilistic views," in *Proc. of the 2005 Intl. Conf. on Very Large Data Bases*, Trondheim, Norway, Aug. 2005.

[23] P. Sen and A. Deshpande, "Representing and querying correlated tuples in probabilistic databases," in *Proc. of the 2007 Intl. Conf. on Data Engineering*, Istanbul, Turkey, Apr. 2007.