Leveraging Spatio-Temporal Redundancy for RFID Data Cleansing

Haiquan Chen
Dept. of CSSE
Auburn University
Auburn, AL, USA
chenhai@auburn.edu

Wei-Shinn Ku*
Dept. of CSSE
Auburn University
Auburn, AL, USA
weishinn@auburn.edu

Haixun Wang
Microsoft Research Asia
Beijing, China
haixunw@microsoft.com

Min-Te Sun
Dept. of CSIE
National Central Univ.
Taoyuan, Taiwan
msun@csie.ncu.edu.tw

ABSTRACT

Radio Frequency Identification (RFID) technologies are used in many applications for data collection. However, raw RFID readings are usually of low quality and may contain many anomalies. An ideal solution for RFID data cleansing should address the following issues. First, in many applications, duplicate readings (by multiple readers simultaneously or by a single reader over a period of time) of the same object are very common. The solution should take advantage of the resulting data redundancy for data cleaning. Second, prior knowledge about the readers and the environment (e.g., prior data distribution, false negative rates of readers) may help improve data quality and remove data anomalies, and a desired solution must be able to quantify the degree of uncertainty based on such knowledge. Third, the solution should take advantage of given constraints in target applications (e.g., the number of objects in a same location cannot exceed a given value) to elevate the accuracy of data cleansing. There are a number of existing RFID data cleansing techniques. However, none of them support all the aforementioned features. In this paper we propose a Bayesian inference based approach for cleaning RFID raw data. Our approach takes full advantage of data redundancy. To capture the likelihood, we design an n-state detection model and formally prove that the 3-state model can maximize the system performance. Moreover, in order to sample from the posterior, we devise a Metropolis-Hastings sampler with Constraints (MH-C), which incorporates constraint management to clean RFID raw data with high efficiency and accuracy. We validate our solution with a common RFID application and demonstrate the advantages of our approach through extensive simulations.

Categories and Subject Descriptors

H.2 [**Information Systems**]: Database Management

General Terms

Algorithms, Design, Experimentation

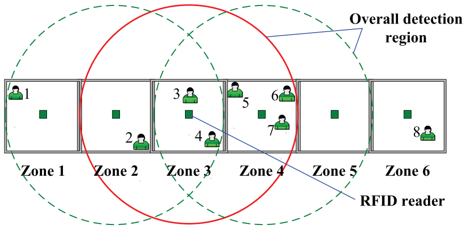


Figure 1: Spatial overlapping of detection regions.

Keywords

Spatio-Temporal Databases, Probabilistic Algorithms, Uncertainty, Data Cleaning

1. INTRODUCTION

Radio Frequency Identification (RFID) is an electronic tagging technology that allows objects to be automatically identified at a distance without a direct line-of-sight, using an electromagnetic challenge/response exchange [27]. An increasing number of major retailers such as Wal-Mart, The Home Depot, Kroger, and Costco have installed RFID based inventory management systems in their warehouses and distribution centers. However, practitioners are facing a challenging problem: the raw data collected by RFID readers are inherently unreliable [23, 17]. Therefore, middleware systems [16] are required to correct readings and provide cleansed data. Most previous solutions [11, 17, 21, 15, 8] for cleansing RFID raw data focused on smoothing the readings generated by a group of readers. However, these existing solutions suffer from three major limitations:

- Data redundancy introduced by overlapping detection regions of multiple stationary readers (spatial redundancy) or continuous readings over time of a single mobile reader (temporal redundancy) is not utilized to improve reading accuracy.

- Prior knowledge about tagged objects and RFID readers is not effectively utilized to improve reading accuracy.

- Constraints in target applications (e.g., the maximal capacity of a room or a shelf) are not effectively utilized to cleanse the data.

In this paper, we propose a method to address these limitations. We focus on taking full advantage of data redundancy, prior knowledge, and application constraints to elevate the accuracy and efficiency of data cleansing.

[*] This research has been funded in part by the National Science Foundation grants CNS-0831502 (CT), CNS-0855251 (CRI).

SIGMOD'10, June 6–11, 2010, Indianapolis, Indiana, USA.

	Zone 1 Reader	Zone 2 Reader	Zone 3 Reader	Zone 4 Reader	Zone 5 Reader	Zone 6 Reader
Obj 1	1	0	0	0	0	0
Obj 2	0	1	1	0	0	0
Obj 3	0	0	0	1	0	0
Obj 4	0	0	1	1	0	0
...	...	...	...	...	...	...

Table 1: RFID readings.

1.1 Data Redundancy

Two types of redundancy may arise in RFID related applications: spatial redundancy, where an object is detected by multiple readers in its neighborhood, and temporal redundancy, where an object is detected multiple times by a single reader over time.

Spatial Redundancy: In order to reduce the complexity of data analysis, previous works [11, 17, 21, 15, 8, 28] assume that each object is read once, and read by one reader only. Clearly, this assumption is difficult to enforce, and more importantly, it oversimplifies the reality. Because RFID readings are of low quality, many applications have to employ redundant readers to cover the target area completely to improve localization accuracy, which means objects are read by multiple readers simultaneously.

Indeed, in RFID systems, spatial redundancy is very common. Figure 1 shows an example of spatial redundancy where the target area is divided into six zones (using one dimensional model) and an RFID reader is located in the center of each zone. Spatial overlap of readers' detection regions leads to duplicate readings, i.e., an object is in the detection regions of multiple readers. A possible set of readings is shown in Table 1 wherein 1's denote successful detections and 0's otherwise. The table shows two effects of redundancy:

- Object 2 is detected by the reader in Zone 2 and also the reader in Zone 3, which makes it difficult to tell the exact location of Object 2. However, since an object cannot appear in more than one zone at the same time, at least one of the readings belongs to spatial redundancy.

- Object 3 is detected in Zone 4 only. However, it does not necessarily mean that Object 3 is in Zone 4 for sure. It is possible that the reader in the zone where Object 3 is located simply failed to detect it.

On the first look, spatial redundancy causes confusion as it introduces inconsistent information (e.g., about the location of Object 2). However, a redundant reading may supply the necessary information for the system to derive the location of an object when its intended reader fails to detect it (e.g., Object 3). Thus, the challenge is how to take advantage of redundancy while avoiding its undesirable effect in data cleansing.

Temporal Redundancy: Besides employing multiple stationary readers, many applications monitor the target area using a mobile reader (e.g., a handheld or a robot-mounted reader [24]) to take continuous readings on its route. Because the exact location of the mobile reader is always changing when the reader reports raw data, the detection regions at different time points may overlap, introducing temporal data redundancy of readings. However, if we treat the same reader at different time points as different readers, e.g., as shown in Table 1, when a mobile reader traverses from zone 1 to zone 6, the reader can be considered as the zone 1 reader while moving in zone 1 and the reader can be treated as the zone 2 reader while moving in zone 2, the temporal redundancy problem can be reduced to the spatial redundancy problem. Therefore, we will mainly focus on spatial redundancy in this paper.

1.2 Prior Knowledge

As false negatives and false positives abound in raw RFID readings [23, 17], in order to recover the true information, the data cleansing system should take prior knowledge into account. Prior knowledge may include information such as, for example, the detection areas of readers in Zone 2 and Zone 3 have significant overlapping, the positioning of the reader in Zone 4 makes it more likely to detect objects in Zone 3 than objects in Zone 5, or the reader in Zone 3 has high false negative rate, etc. Such information, when properly integrated with the readings, is extremely valuable for data cleansing.

1.3 Constraints

Environmental constraints can be utilized to improve data cleansing. For example, the maximal capacity of each zone (the number of objects that can reside in the same zone) is a constraint. If each zone represents a rack or shelf in a warehouse, one possible constraint is the total size or weight of the objects which the rack can hold. In addition to these physical constraints, information obtained from other channels can be translated into constraints. For instance, if an extra source indicates that two certain objects are in the same zone, it may help cleanse the data of these two as well as other objects when the information is integrated with readings and other constraints.

1.4 Overview of Our Approach

In this paper, we propose an innovative approach of cleansing RFID raw data which is able to take full advantage of duplicate readings and integrate prior knowledge as well as environmental constraints. Our approach is based on Bayesian inference. We introduce an n-state detection model and prove by entropy analysis that the 3-state detection model can maximize the system performance. Furthermore, we devise a Metropolis-Hastings sampler with constraints to efficiently approximate the posterior (Metropolis-Hastings sampling is a Markov Chain Monte Carlo method [22, 2]). Consequently, our approach enables two important types of queries against RFID raw data: the location query and the aggregate query (e.g., about remaining capacity of each zone). The contributions of this study are as follows:

- By using Bayesian inference, we derive a universal framework for computing the posterior probabilities (of the location of each object).

- Based on the physical characteristics of RFID readers, we propose an n-state detection model to capture likelihoods, which enables us to take full advantage of duplicate readings.

- We analyze the relationships between the system entropy and the read rate of RFID readers under the 2-state and 3-state detection models, respectively.

- By investigating the impact of the number of states in a detection model on the system entropy, we formally prove that the system entropy can be minimized if the 3-state model is adopted compared with other state models. In other words, having even more states (greater than 3) can in fact deteriorate the overall system performance.

- We devise MH-C, an improved Metropolis-Hastings sampler, to sample from the posterior while taking the environmental constraints into consideration.

Symbol	Meaning
$\hat{H}$	The random vector representing locations of all objects
h_i	The random variable representing the location of o_i
$\mathbb{Z}$	Raw data reported by RFID readers
z_{ij}	The raw data (0 or 1) reported by the reader in zone j for object o_i
$post(\hat{H}\|\mathbb{Z})$	The posterior probability of the location vector $\hat{H}$ given the raw data $\mathbb{Z}$
$p(z_{ij}\|h_i)$	The likelihood that the zone j reader reports the value of z_{ij} for object o_i given that object o_i is in the zone h_i
$p(h_j)$	The prior probability that object o_j is in the zone h_j

Table 2: Symbolic notations of Section 2.

- We demonstrate the efficiency and effectiveness of our approach by comparing the performance of MH-C with the Sequential Importance Sampling (SIS) based solution [28] through extensive simulations.

1.5 Paper Organization

The rest of this paper is organized as follows. The Bayesian inference-based framework of our approach is presented in Section 2. In Section 3, we propose the n-state detection model to take full advantage of duplicate readings. We prove that the 3-state detection model can maximize the system performance in Section 4. In Section 5, we introduce a Metropolis-Hastings sampler with constraints. The experimental validation of our design is presented in Section 6. Section 7 surveys the related work. Finally, Section 8 concludes this paper.

2. BAYESIAN INFERENCE

In this section, we develop a Bayesian inference-based approach to handle redundant readings and prior knowledge, and we analyze the challenges when applying this approach. Table 2 summarizes the notations used in this section.

2.1 A Bayesian Inference Based Approach

Bayesian inference is a statistical inference technique that estimates the probability of a hypothesis (x) based on observations (y). Bayesian inference shows that posterior is proportional to the multiplication of likelihood and prior, which can be represented as $p(x|y) \propto p(y|x)p(x)$.

Suppose there are m zones and n objects in our monitoring environment, each zone with a reader mounted in the zone center. Let o_i represent the object with ID i. For each o_i, its location is represented by a random variable h_i. Therefore, a possible distribution of n objects in m zones can be denoted as an instance of the random vector $\hat{H} = (h_1, h_2, \ldots, h_n)$. h_i represents the zone ID where object o_i is in. For example h_1 = 2 denotes that object o_1 is in zone 2 in the current instance. For the reader in zone j, the raw data (0 or 1) it receives from the RFID tag of object o_i is denoted as z_{ij}. The raw data matrix for each complete scan from m readers can then be represented as an $n \times m$ matrix $\mathbb{Z} = [z_{ij}]$. Thus the Bayes' theorem can be represented as Equation 1, where $post(\hat{H}|\mathbb{Z})$ denotes the posterior probability of location vector $\hat{H}$ given the raw data $\mathbb{Z}$, and a valid hypothesis means the hypothesis satisfies all constraints:

$$\begin{array}{ll} post(\hat{H}|\mathbb{Z}) = 0 & : \hat{H} \text{ is not valid} \\ post(\hat{H}|\mathbb{Z}) > 0 & : \hat{H} \text{ is valid} \\ post(\hat{H}_1|\mathbb{Z}) > post(\hat{H}_2|\mathbb{Z}) & : \hat{H}_1 \text{ is more likely than } \hat{H}_2 \end{array}$$

In particular, if z_{ij} = 1 in a raw data matrix and the actual location of object o_i is not in zone j, then z_{ij} is a false positive. Take Table 1 as an example, at least one of z_{22} and z_{23} is a false positive because object 2 cannot be in zone 2 and zone 3 simultaneously. Similarly, at least one of z_{43} and z_{44} is a false positive.

To compute $post(\hat{H}|\mathbb{Z})$, we make some independence assumptions of random variables. RFID reader transmissions or tag transmissions may lead to collisions because readers and tags communicate over a shared wireless channel. Reader collisions happen when neighboring readers communicate with a tag simultaneously [9] and tag collisions occur when multiple tags transmit to a reader at the same time [10]. However, the two kinds of collisions can be effectively prevented by arbitration protocols (e.g. by scheduling adjacent readers to operate at different times) [25, 13, 20]. Therefore, we assume each reader detects the tags of different objects independently (i.e., whether a reader can successfully detect the tag of a certain object does not interfere with whether the reader can successfully detect that of another object) in this research. Based on the assumption, we can derive Equation 2.

Furthermore, because we employ MH-C to take into account constraints (i.e., to ensure that each generated sample satisfies all the constraints), here we can simply assume independence between different h_i (i.e., the locations of objects). In addition, we assume that each reader's detection of the same object is independent. Besides, the prior distribution of each object does not depend on that of other objects. Therefore, we can obtain Equation 3. If we rewrite Equation 3 using the normalizing constant, denoted as α, we can reach Equation 4, which shows how to compute the posterior of each sample. To be specific, $p(z_{ij}|h_i)$ reflects the corresponding *likelihood*, which is the probability that the reader in zone j reports the value of z_{ij} about object o_i given that object o_i is actually in the zone with ID h_i. Furthermore, $p(h_j)$ denotes the *prior probability* that the object o_j is in the zone with the ID of h_j. The prior probability can be interpreted as the assumed distribution before acquiring the RFID raw data.

2.2 The Goal and the Obstacles

Based on Equation 4, given the raw readings $\mathbb{Z}$ and a hypothesis $\hat{H}$ (the location of each object), we can derive the probability of the hypothesis. However, finding just one valid hypothesis will provide nothing more than a biased answer to queries against the uncertain data. To address this issue, we need to query against all valid hypotheses. However, this is unrealistic because there are numerous valid hypotheses in most cases. Thus, our goal is to create a large sample set of valid hypotheses, each associated with a weight computed by Equation 4: $(\hat{H}_1, w_1), (\hat{H}_2, w_2), \cdots, (\hat{H}_n, w_n)$. The sample set of valid hypotheses as a whole enables us to answer queries with high credibility. To achieve this goal, we must overcome the following obstacles:

- A prerequisite for effective hypothesis sampling is to be able to compute the posterior probability of each hypothesis precisely. Therefore, we propose the n-state detection model in Section 3 to capture likelihoods in an affordable and accurate way.

- The hypothesis space is high dimensional, and the posterior probability is difficult to sample from. Thus, we need a sampling technique that has desirable efficiency. In Section 5, we apply a Markov Chain Monte Carlo method (MCMC) because MCMC can maintain the correlation between samples, resulting in an improved sampling efficiency.

- We need to incorporate constraint management in sampling. We propose a sampler called Metropolis-Hastings sampler with Constraints (MH-C), which improves the naive Metropolis-Hastings (MH) sampler. Each sample generated by MH-C automatically satisfies all the constraints.

$$\begin{aligned} post(\hat{H}|\mathbb{Z}) &= post(h_1, h_2, \ldots, h_n | \begin{bmatrix} z_{11} & \cdots & z_{1m} \\ \vdots & \ddots & \vdots \\ z_{n1} & \cdots & z_{nm} \end{bmatrix}) \\ &\propto p(\begin{bmatrix} z_{11} & \cdots & z_{1m} \\ \vdots & \ddots & \vdots \\ z_{n1} & \cdots & z_{nm} \end{bmatrix} | h_1, h_2, \ldots, h_n) \cdot p(h_1, h_2, \ldots, h_n) \end{aligned} \tag{1}$$

$$post(\hat{H}|\mathbb{Z}) \propto \prod_i p(z_{i1}, z_{i2}, \ldots, z_{im} | h_1, h_2, \ldots, h_n) \cdot p(h_1, h_2, \ldots, h_n) \tag{2}$$

$$post(\hat{H}|\mathbb{Z}) \propto \prod_i p(z_{i1}|h_i) \cdot p(z_{i2}|h_i) \cdot \ldots \cdot p(z_{im}|h_i) \cdot \prod_j p(h_j) \tag{3}$$

$$post(\hat{H}|\mathbb{Z}) = \alpha \cdot \prod_i p(z_{i1}|h_i) \cdot p(z_{i2}|h_i) \cdot \ldots \cdot p(z_{im}|h_i) \cdot \prod_j p(h_j) \tag{4}$$

3. RFID READER DETECTION MODELS

The major difficulty in computing the posterior of each sample (Equation 4) lies in how to accurately estimate the likelihood $p(z_{ij}|h_i)$. To do so, we introduce the n-state detection model to capture likelihood in an affordable and precise way.

3.1 Physical Characteristics

RFID data acquisition and transmission are unreliable [10, 15, 17, 21]. In our experiment, we investigated the change of the read rate over distance using regular RFID readers and tags. The results are illustrated in Figure 2. The model of the tags is Gen2 RFID Smart Label and the model of the reader is MPR-6000 (antenna 902-928 MH), provided by WJ Communications Inc. Our test environment is a lab with many metal objects (tables, desks and computer equipment), representing a noisy environment.

As Shown in Figure 2, the overall detection range of a reader can be separated into the major detection region and the minor detection region, where in the major detection region from 0 to almost 5 feet, the read rate can keep a level of around 95% and in the minor detection region approximately from 5 to 13 feet, the read rate drops off almost linearly. Furthermore, the read rate deteriorates to zero in the region more than 13 feet away from the reader, which is defined as beyond the overall detection range [17].

3.2 Problems of the 2-State Detection Model

Taking the scenario in Figure 1 as an example, one way to estimate likelihood is as follows:

$$p(z_{ij} = 1|h_i) = \begin{cases} r & \text{if } h_i \in \{j-1, j, j+1\} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$p(z_{ij} = 0|h_i) = \begin{cases} 1-r & \text{if } h_i \in \{j-1, j, j+1\} \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

where r is the average read rate. Intuitively, it means that an object o_i holds the same probability to be detected by a reader whether o_i is in the zone (j) the reader is associated to, or in any of neighboring zones ($j-1$ or $j+1$). Apparently, if compared with Figure 2, this 2-state detection model is inherently inaccurate as it fails to capture any change of the read rate in the overall detection region. Consequently, when predicting the location of an object, the resulting system is unable to differentiate between its own zone and all its neighboring zones because all the above zones are with an identical read rate.

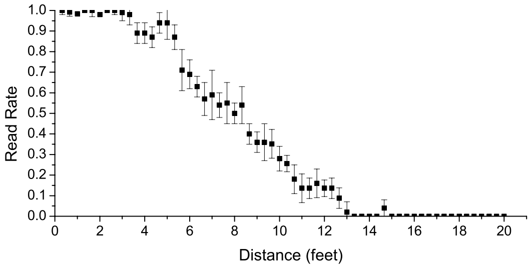


Figure 2: An illustration of the relationship between read rate and distance.

In order to solve this problem, current works are forced to adopt a simplified 2-state detection model, which is shown in Figure 3. This simplified 2-state model assumes readers' detection regions do not overlap, i.e., a reader is only able to detect the objects in its own zone. Then the likelihood can be estimated as follows:

$$p(z_{ij} = 1|h_i) = \begin{cases} r & \text{if } h_i = j \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$p(z_{ij} = 0|h_i) = \begin{cases} 1-r & \text{if } h_i = j \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

This simplified 2-state model, however, has two problems. First, it is unrealistic to assume that we can divide the space into non-overlapping detection regions. Second, the model does not support applications that use redundant information (such as redundant readings) to offset the unreliability of raw RFID readings.

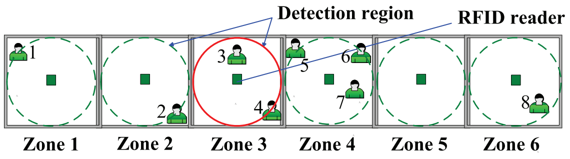


Figure 3: An illustration of the simplified 2-state model.

3.3 The n-state Detection Model

In order to take full advantage of duplicate readings, we propose an n-state detection model, which is illustrated in Figure 4. In Figure 4, the overall detection region of an RFID reader is divided into several sub-regions, each of which corresponds to a zone associated with a unique read rate. As far as a specific detection model is concerned, the difference in the read rate over any two adjacent sub-regions is a constant, i.e., the read rates for different states constitute an arithmetic sequence. Take the 4-state model as an example. Suppose the highest read rate in the model is x. The first state (counted with the increase of the detection distance) holds a read rate of x, the second state keeps a read rate of $\frac{2x}{3}$, the third state maintains a read rate of $\frac{x}{3}$, and the fourth state eventually has a read rate of zero. Thus, as for a specific reader, by employing the n-state detection model, each correlated zone is assigned a distinct read rate according to its distance to this reader. In particular, the simplest model in the family of the n-state detection model is the 2-state model, where an identical read rate is assumed in the overall detection region of each reader.

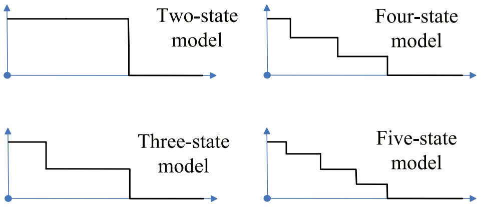


Figure 4: The family of the n-state detection model.

Notice that in practice, the n value depends on how zones are divided in overall detection regions of RFID readers. This is because an n-state model in fact implies that every $2(n-2)+1 = 2n-3$ zones correlate with each other (assuming that all the zones are in a 1-dimensional distribution). For example, if it is known as prior knowledge that one object can be read simultaneously by up to five readers, we have to choose $n = 4$ to incorporate the correlation among every 5 successive zones. A formal derivation of the location distribution in terms of the probabilistic mass function can be found in Section 4.

3.4 A Case Study: The 3-State Model

We elaborate on the 3-state model as a case study. Suppose one reader can only detect its own zone and the two neighboring zones. This assumption implies that, as for a particular reader, there are three distinct location-based states of a object: in the same zone as the reader, in the neighboring zones, and in all the other zones. To capture this correlation, we have to choose $n = 3$ and the resulting model is the *3-state detection model*, where the overall detection range of a reader is divided into two sub-regions, as shown in Figure 5. Specifically, the major detection region, the minor detection region and the zero read rate region in Figure 5 correspond to the zone where the reader locates, neighboring zones and all the other zones, respectively. Therefore, the motivating scenario (Figure 1), if interpreted by the 3-state model, can be illustrated in Figure 6.

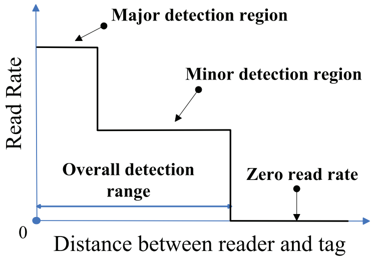


Figure 5: The 3-state detection model of RFID readers.

In Figure 6, by using the 3-state detection model, not only the duplicate readings can be incorporated, but also a zone and all its neighboring zone can be differentiated because they are with distinct read rates. To be specific, if object o_i is in zone j, not only z_{ij} but also $z_{i(j-1)}$ and $z_{i(j+1)}$ should have a considerable chance to be 1 (false positives). In the meantime, other readers are unable to detect the tag of object o_i. The reason is that object o_i may be in the major detection region of the reader in zone j while it may be also in the minor detection region of both readers in zones $j-1$ and $j+1$. As for the other readers, object o_i is totally beyond their overall detection regions, leading those readers to report 0 for object o_i. If we denote the mean value of the read rate in major detection region as r_{major} and the mean value of the read rate in minor detection region as r_{minor}, the estimate of the likelihood using the 3-state model can be represented in Equation 9 and Equation 10.

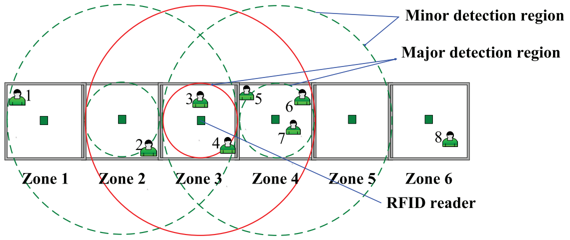


Figure 6: The detection-region overlap interpreted by the 3-state detection model.

$$p(z_{ij} = 1|h_i) = \begin{cases} r_{major} & \text{if } h_i = j \\ r_{minor} & \text{if } h_i \in \{j-1, j+1\} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$p(z_{ij} = 0|h_i) = \begin{cases} 1 - r_{major} & \text{if } h_i = j \\ 1 - r_{minor} & \text{if } h_i \in \{j-1, j+1\} \\ 1 & \text{otherwise} \end{cases} \tag{10}$$

The advantage of the 3-state detection model over the 2-state detection model can be measured by the system entropy (Section 4). We also answer the question whether having even more states (more than 3) can further benefit the system.

4. ENTROPY ANALYSIS

We use entropy to measure the uncertainty in a system after invalid system states have been eliminated by a data cleansing method. Generally, applying an efficient data cleansing method will lead to systems with smaller entropy. In this section, we firstly show the advantage of the 3-state model over the 2-state model and then prove that the 3-state model can maximize the system performance compared with other detection models with even more than 3 states. A snippet of the RFID raw data is shown in Table 3 and the actual location of an object i is denoted as a random variable L.

4.1 Entropy versus Read Rate

Entropy of the 2-State model: Suppose that y denotes the read rate in the 2-state detection model. According to the right side of Equation 4, the probabilistic mass function of L in the 2-state model can be represented as:

$$p(L = l) = \begin{cases} \alpha(1-y)y(1-y)\beta & \text{if } l = j \\ \alpha(1-y)(1-y)y\beta & \text{if } l \in \{j-1, j+1\} \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where α is the normalizing constant and β represents the prior probability (we assume the prior distribution as a uniform distribution) in Equation 4. Thus, we can calculate the entropy of the

	...	Zone $j-1$ Reader	Zone j Reader	Zone $j+1$ Reader	...
...	...	...	...	...	...
Obj i	...	0	1	0	...
...	...	...	...	...	...

Table 3: A snippet of the RFID raw data.

distribution of L as:

$$\begin{aligned} H(L) = \quad & -\alpha(1-y)(1-y)y\beta \cdot \ln(\alpha(1-y)(1-y)y\beta) \quad (12)\\ & -\alpha(1-y)y(1-y)\beta \cdot \ln(\alpha(1-y)y(1-y)\beta) \\ & -\alpha(1-y)(1-y)y\beta \cdot \ln(\alpha(1-y)(1-y)y\beta) \end{aligned}$$

Because the probabilities on all the locations sum to 1, we can derive Equation 13. By applying Equation 13 to Equation 12 (α and β are canceled out), we can obtain Equation 14.

$$\alpha\beta = \frac{1}{3(1-y)^2 y} \tag{13}$$

$$H(L) = -3 \cdot \frac{1}{3} \cdot \ln \frac{1}{3} = 1.098 \tag{14}$$

Entropy of the 3-State model: Figure 6 corresponds to the 3-state model scenario. Suppose x is the read rate in the major detection region. Then the read rate in the minor detection region can be denoted as $x/2$. Thus, according to the right side of Equation 4, the probabilistic mass function of L can be represented as follows:

$$p(L=l) = \begin{cases} \alpha(1-\frac{x}{2})x(1-\frac{x}{2})\beta & \text{if } l = j \\ \alpha(1-\frac{x}{2})(1-x)\frac{x}{2}\beta & \text{if } l \in \{j-1, j+1\} \\ 0 & \text{otherwise} \end{cases}$$

Similarly, α is the normalizing constant and β represents the prior probability in Equation 4. Therefore, we can calculate the entropy of the distribution of L as:

$$\begin{aligned} H(L) = \quad & -\alpha(1-\frac{x}{2})(1-x)\frac{x}{2}\beta \cdot \ln(\alpha(1-\frac{x}{2})(1-x)\frac{x}{2}\beta) \\ & -\alpha x(1-\frac{x}{2})^2\beta \cdot \ln(\alpha x(1-\frac{x}{2})^2\beta) \quad (15) \\ & -\alpha(1-\frac{x}{2})(1-x)\frac{x}{2}\beta \cdot \ln(\alpha(1-\frac{x}{2})(1-x)\frac{x}{2}\beta) \end{aligned}$$

Since probabilities on all locations sum to 1, we can obtain Equation 16.

$$\alpha\beta = \frac{1}{x(1-\frac{x}{2})(2-\frac{3x}{2})} \tag{16}$$

Combining Equation 16 and Equation 15, we have:

$$H(L) = -2 \cdot \frac{1-x}{4-3x} \cdot \ln \frac{1-x}{4-3x} - \frac{2-x}{4-3x} \cdot \ln \frac{2-x}{4-3x}$$

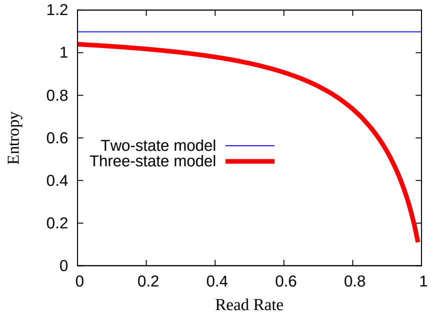


Figure 7: Relationship between entropy and read rate in the 2-state and 3-state models.

In Figure 7, we plot the relationship between the reconstruction entropy and read rate under the 2-state and 3-state models, respectively. As Figure 7 illustrates, the entropy will decrease accordingly with the increase of read rate, which indicates that the system will have less uncertainty with more reliable readers. Moreover, Figure 7 shows that the entropy in the 3-state model is always smaller than that in the 2-state model. For example, if $x = 0.95$, the entropy in the 3-state model is 0.395 while the entropy in the 2-state model is 1.098. This observation reveals that the 3-state model can be more informative in object localization than the 2-state model.

4.2 Entropy versus Number of States

Here we investigate the relationship between system entropy and the number of states in a detection model. Suppose an n-state model is adopted with the highest read rate of x. Thus, the read rate in the i^{th} state (counted with the increase of the detection distance), can be represented as $\frac{(n-i)\cdot x}{n-1}$. Combined with Equation 4 and Table 3, we can obtain the probabilistic mass function of L, represented as Equation 17. Equation 17 shows that in an n-state model, a successful reading "1" of a certain reader about an object in fact implies that this object may exist in any of the $2(n-2)+1 = 2n-3$ correlated zones (including the zone which the reader is associated to), each with a non-zero probability.

Based on Equation 17, we plotted the relationship between entropy and the number of states in a detection model in Figure 8, where we assume x equals to 0.95 (the most common case). According to Figure 8, the 3-state detection model can minimize the system entropy and lead to the maximum system performance. In other words, having more states (more than 3) in a detection model can even deteriorate the system performance. Therefore, our experiments are mainly focused on the 3-state model.

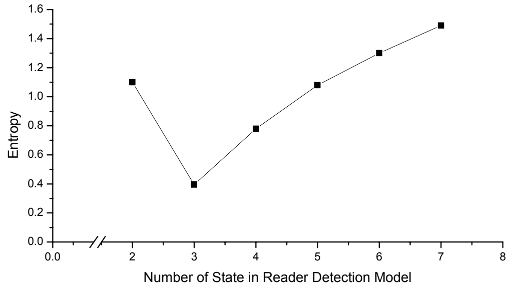


Figure 8: Relationship between entropy and the number of states in a detection model.

5. SAMPLING

By using Bayesian inference, we derive the posterior, as shown in Equation 4. Since Equation 4 is easy to compute but hard to sample from, we need an efficient method to draw samples from the posterior distribution. In this section, we briefly review Markov Chain Monte Carlo (MCMC) and then show why MCMC is chosen in our solution. Next, as the two most commonly used MCMC samplers, the difference between the Metropolis-Hastings sampler and the Gibbs sampler is discussed. Finally, we propose a Metropolis-Hastings sampler with Constraints (MH-C) method.

5.1 Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) is a technique to generate samples from the state space by simulating a Markov chain. The formed Markov chain is constructed in such a way that the chain spends more time in the regions with higher importance (i.e., the Markov chain converges to the posterior as its stationary distribution). As the number of samples is sufficiently large, all the samples

$$p(L=l)=\begin{cases}\alpha(1-\frac{x}{n-1})(1-\frac{2x}{n-1})...(1-\frac{(n-2)x}{n-1})\cdot x\cdot(1-\frac{(n-2)x}{n-1})...(1-\frac{2x}{n-1})(1-\frac{x}{n-1})\beta & \text{if } l=j\\ \alpha(1-\frac{x}{n-1})(1-\frac{2x}{n-1})...(1-\frac{(n-2)x}{n-1})\cdot\frac{(n-1-k)x}{n-1}\cdot(1-\frac{(n-2)x}{n-1})... & \\ (1-\frac{(n-1-k+1)x}{n-1})\cdot(1-x)\cdot(1-\frac{(n-1-k-1)x}{n-1})...(1-\frac{2x}{n-1})(1-\frac{x}{n-1})\beta & \text{if } l\in\{j-k,j+k\},k\in\{1,2,...,n-2\}\\ 0 & \text{otherwise}\end{cases} \tag{17}$$

become the fair samples from the posterior. Consequently, we are able to approximate a sophisticated posterior based on deliberately constructing such a Markov Chain of samples.

5.2 Sample Correlation

In our design, we choose MCMC instead of other sampling technique because MCMC maintains the correlation among samples. In MCMC, the next sample depends on the current sample. Before we elaborate on how we can take advantage of sample correlation to improve the efficiency of sampling in our scenario, we define two terms as follows.

Definition We call any sample generated by the sampler a *candidate sample*. A *qualified sample* is a candidate sample that satisfies all constraints.

The existence of constraints leads to the uniqueness of our problem. Samples must satisfy all the constraints to become qualified. Note that in most sampling problems, we prefer independent samples, that is, the current draw of a sample is independent from the previous draw. In our scenario, however, the sampling techniques which generate independent samples (e.g., importance sampling) may suffer from low sampling efficiency due to the loss of correlation between adjacent samples. Figure 9 illustrates how the correlation between samples can be utilized to improve the sampling efficiency. The qualified sampling space is a subset of the complete sampling space. Suppose sample point A is the current sample in the qualified sampling space. For an independent sampler, the next sample could be any point in the complete sampling space. However, the next sample is useful only if it happens to fall into the qualified sampling space. On the contrary, if a MCMC-based sampler is employed, the next sample will be chosen according to the proposal distribution at point A, i.e., the next sample will be in the area denoted by the dotted circle centering at point A. Therefore, compared to other independent samplers, the probability that the next sample generated by MCMC falls into the qualified sampling space is considerably increased.

Note that although MCMC improves sampling efficiency, a sample generated by MCMC may not necessarily be a qualified sample. As in Figure 9, point B is a sample generated by MCMC after point A. However, point B is outside the qualified sampling space. Consequently, we have to sample again to acquire point C, which is a qualified sample, and then add it into the Markov chain as the next state (i.e., the Markov chain moves from point A to point C).

5.3 Metropolis-Hastings and Gibbs Sampling

The Metropolis-Hastings (MH) sampler and the Gibbs sampler are the two most common MCMC samplers. MH conducts a sequence of random walks using a proposal distribution and decides whether to reject the proposed moves using the rejection sampling. In the applications of Bayesian inference, the normalizing constant is usually extremely difficult to compute. MH avoids the computation of the constant. It approximates the posterior by using only the ratio of the posterior, where the constant is canceled out.

Recall that the random vector representing the locations of objects is denoted as $\hat{H}$ and the posterior distribution is $post(\hat{H}|\mathbb{Z})$.

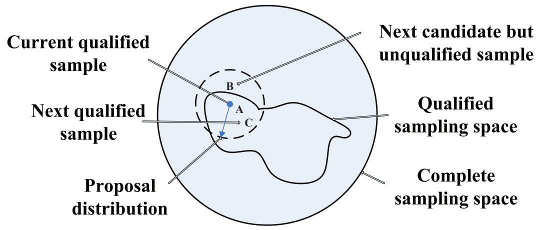


Figure 9: Taking advantage of the correlation between samples to improve sampling efficiency.

Suppose $\hat{H}_{t-1}$ is the immediate previous state before the state $\hat{H}_t$ in the formed Markov chain. According to the MH algorithm, at first, a proposal sample, $\hat{H}_q$, is drawn from a proposal distribution, $q(\hat{H}_q|\hat{H}_{t-1})$, i.e., $\hat{H}_q$ is a random deviation from $\hat{H}_{t-1}$. In our research, we use a uniform proposal distribution whose support is defined as the step length. The proposal sample $\hat{H}'$ can be denoted as $\hat{H}_{t-1}$+ $\hat{H}_q$. Then MH accepts $\hat{H}'$ as the next state $\hat{H}_t$ with the probability of $\frac{post(\hat{H}'|\mathbb{Z})}{post(\hat{H}_{t-1}|\mathbb{Z})}$.

Here we compare MH sampler with the Gibbs sampler in brief. The Gibbs sampler requires that conditional (marginal) distributions for each variable are known and easy to sample from. MH relies on the ratio of the posterior, and does not require to sample from any distribution. Because we have already derived the closed form of the posterior as Equation 4 and are able to calculate likelihoods easily according to the proposed n-state model, it will be much more straightforward to use MH sampler rather than the Gibbs sampler in our design.

5.4 MH-C

Although the naive MH algorithm can evaluate the posterior by forming a Markov chain in the sampling space, it does not take constraints into account. If we impose constraints to samples, many of them should be rejected because they are inapplicable, i.e., they are not qualified samples.

To incorporate constraints in sampling, we propose a Metropolis-Hastings sampler with Constraints (MH-C). With MH-C, each zone is associated with multiple variables called *resource descriptors*. The current value of a *resource descriptor* represents how much the associated resource is still available. Suppose we have a variable, denoted as $Descriptor_{zone_i}$, to keep track of the current available vacancy in zone i. The initial value of $Descriptor_{zone_i}$ is set to the maximal capacity of zone i. Thus, whether an object j with the volume $Volume_{object_j}$ is able to be stored in zone i can be examined by:

$$Descriptor_{zone_i} = Descriptor_{zone_i} - Volume_{object_j} \tag{18}$$

The proposed resource allocation is feasible only if $Descriptor_{zone_i}$ is no less than zero. Otherwise, we have to re-sample until a new allocation meets all the constraints. Consequently, the problem of whether an allocation is feasible (or compatible) can reduce to the problem of monitoring the value of each descriptor.

With MH-C, because each sample is a D_{object}-dimensional vector, a proposal sample is generated iteratively dimension by dimen-

Symbol	Meaning
$\mathbb{Z}$	The raw data matrix from RFID readers
$\mathbb{S}$	The sample set
$\vec{C}$	The current sample in the Markov chain
$\vec{P}$	The proposal sample in the Markov chain
C_j	The j^{th} dimension of $\vec{C}$
P_j	The j^{th} dimension of $\vec{P}$
E	The number of effective samples
B	The number of samples in the burn-in phase
S	The step length for the uniform proposal distribution
D_{object}	The total number of monitored objects
D_{zone}	The total number of zones
$Jitter$	A random number between 0 and 1
$Rand(a, b)$	Generate a random integer between a and b based on uniform distribution
$Post(\hat{H}\|\mathbb{Z})$	The posterior probability of the sample $\hat{H}$ given raw data $\mathbb{Z}$

Table 4: Symbolic notations used in Algorithm 1.

sion. If any descriptor for the current allocation is less than zero, there will be no chance for the current partial sample to become a qualified sample. Therefore, we can discard the current value and then choose another value for that dimension by re-sampling. As far as the proposal distribution is concerned, we construct a random walk chain by choosing a uniform proposal distribution within the step length. A detailed description of MH-C algorithm is illustrated in Algorithm 1 and the related notations are summarized in Table 4.

In Algorithm 1, line 1 initializes the sample set and takes the RFID raw data. Line 2 loads the n-state detection model of readers with the objective of computing the likelihood. Line 3 initializes all the resource descriptors and line 4 randomly chooses a qualified sample as the first state of the Markov Chain. Lines 6 to 17 generate a random sample as the proposal sample dimension by dimension (object by object). Lines 8 to 13 correspond to the sampling process. First, we obtain a random integer based on the current value and the random proposal value. The correct range of the value on each dimension of the sample vector, represented as h_i, is [1, D_{zone}]. Therefore, if the proposal value P_j overflows in the range, we need to make the value reflect into the range. Then, we check all the related descriptors to make sure their updated values are no less than zero. If any value is less than zero, it means that the current allocation will violate the corresponding constraints. Consequently, we go back to line 8 to re-sample until an allocation on that dimension is feasible, as shown in line 15. Note that our algorithm guarantees each proposal sample is also a qualified sample. After a complete proposal sample is generated, lines 18 to 21 accept this proposal sample as the next state of the Markov chain with the probability of the posterior ratio of the proposal sample over the current sample. Line 22 adds the next state into the sample set. Line 23 resets all the resource descriptors to make sure that the examination on descriptors for the next proposal sample is correct. Note that line 5 is to set the upper bound of the sampling loop to E + B in order to guarantee that the final number of samples in sample set is E + B. It is because the first B samples should be excluded as burn-in samples and consequently only the remaining E samples will be taken into account to reconstruct the posterior.

6. EXPERIMENTAL VALIDATION

In this section, we applied our method to a warehouse application, i.e., each object corresponds to a case and each zone corresponds to a rack. To capture the likelihood, without loss of generality, we assumed that the 3-state detection model was adopted in this application. Also, we implemented MH-C (we use the terms MH-C and MCMC interchangeably in this section) to sample from the posterior. For comparison with our MCMC-based solution, we extended the SIS-based approach in [28] to incorporate duplicate readings because their approach does not consider duplicate readings and only focuses on the distribution of the missing cases.

Algorithm 1 Metropolis-Hastings Sampler with Constraints

```
1: Set $\mathbb{S}$ = $\emptyset$ and take raw data $\mathbb{Z}$
2: Load the $n$-state detection model
3: Initialize all the resource descriptors to their maximal capacity.
4: Initialize $\vec{C}$ by randomly choosing a qualified sample within the support
   of $Post(\hat{H}|\mathbb{Z})$ as the starting point.
5: for $Cycle$ = 2 to $E$+$B$ do
6:   for $j$ = 1 to $D_{object}$ do
7:     repeat
8:       $P_j$ = $C_j$+ Rand(-$S$,$S$)
         {Generate a new integer based on the current value and a pro-
         posal value within the step length}
9:       if $P_j < 1$ then
10:        $P_j$ = 1+ (1-$P_j$)
           {Overflow and Reflection}
11:      end if
12:      if $P_j > D_{zone}$ then
13:        $P_j$ = $D_{zone}$-($P_j$-$D_{zone}$)
           {Overflow and Reflection}
14:      end if
15:    until The value of any resource descriptor related to the referred
       zone is no less than zero after the proposed allocation on the cur-
       rent object is committed
16:    $j \leftarrow j + 1$
17:  end for
18:  Generate a random number between 0 and 1: $Jitter$
19:  if $Jitter \leq \min(1, \frac{Post(\vec{P}|\mathbb{Z})}{Post(\vec{C}|\mathbb{Z})})$ then
20:    $\vec{C}$=$\vec{P}$
       {Metropolis-Hastings}
21:  end if
22:  Add $\vec{C}$ into $\mathbb{S}$ as the next sample
23:  Resetting all the resource descriptors
24:  $Cycle \leftarrow Cycle + 1$
25: end for
```

Parameter	**Small-scale setting**	**Large-scale setting**
D_{object}	20	5000
D_{zone}	5	200
B	50	50
S	1	30
$Volume_{object}$	1	1
$Capacity_{zone}$	7	50

Table 5: The parameters for our simulations.

Our simulations are based on two settings, as shown in Table 5. In order to show the scalability of our approach, we employed the large-scale warehouse setting to generate randomly the true distribution matrix and the corresponding noisy RFID raw matrix 100 times to investigate the advantage of our MCMC-based approach over the SIS-based approach [28] in terms of reconstruction efficiency and accuracy. On the other hand, in order to show query results returned by MCMC and SIS, we conducted experiments on a specific true distribution matrix and noisy raw matrix, as shown in Table 6, based on the small-scale warehouse setting.

6.1 Simulator Implementation

Our simulator consists of seven components as displayed in Figure 10. The true matrix generator randomly produces distribution matrices as true distributions. The rows represent cases (objects) and columns represent racks (zones). On the contrary, noisy matrix generator provides the noisy matrices as the RFID raw data in the same format. Then MCMC and SIS modules reconstruct the distribution for each case using the noisy matrix as the input. Our simulator generates the synthetic RFID raw data with the duplicate readings according to the physical characteristics of RFID readers [17]. The 3-state detection model was used to capture the likelihood. Also, we assume as the prior distribution that each case exists on each rack with the same probability. All our experiments were conducted on a Linux machine with an Intel Pentium 4 2.4GHz processor with 2GB of memory.

We employed K-L divergence, the top-1 success rate, and the top-2 success rate to evaluate the reconstruction accuracy. Specifically, K-L divergence is a metric commonly used to evaluate the difference between two distributions. In our research, we calculated the K-L divergence from the reconstructed distribution to the true distribution, i.e., the smaller value of K-L convergence indicates the higher accuracy of the reconstructed distribution. The recovered matrices (reconstructed distributions) were inverted by the matrix inversion module to facilitate the computation of K-L divergence. Then the K-L divergence module was used to compare the reconstructed distributions with the true distributions and compute the corresponding K-L divergence values for MCMC and SIS. On the other hand, the top-k success rate reflected how many cases can be located precisely at a certain resolution using data cleansing techniques. The top-k analysis module was responsible for calculating the top-k success rate.

Definition The *top-k success rate* is a percentage of the number of cases whose true locations match the top k predicted locations of the reconstructed distribution over the total number of cases.

6.2 The Performance Analysis of MH-C

In this section we focus on the performance of MCMC and SIS with respect to reconstruction efficiency and accuracy in the large-scale warehouse setting where there are 5000 cases and 200 racks in total as shown in Table 5. In the reconstruction accuracy experiment, we randomly generated the true distribution matrix and the corresponding RFID noisy matrix 100 times. During each distribution reconstruction, we recorded the sampling time, and computed the average K-L divergence, the top-1 success rate and the top-2 success rate of all the 5000 cases (i.e., each result in this subsection is obtained by averaging over 5000 location queries).

6.2.1 The Reconstruction Efficiency

Here we investigated the performance of MCMC and SIS in terms of the average sampling time. Compared to SIS, the average sampling time of MCMC is remarkably reduced over different number of qualified samples as illustrated in Figure 11. For example, with 5000 qualified samples the sampling time of MCMC is 11.58 seconds while the sampling time of SIS is 230.78 seconds. This is because MCMC takes advantage of the current qualified sample to generate the next qualified sample (i.e., keeping the relevance of samples). Consequently, MCMC takes less time than SIS to come up with the same number of qualified samples .

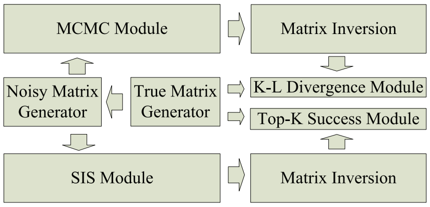


Figure 10: The simulator structure.

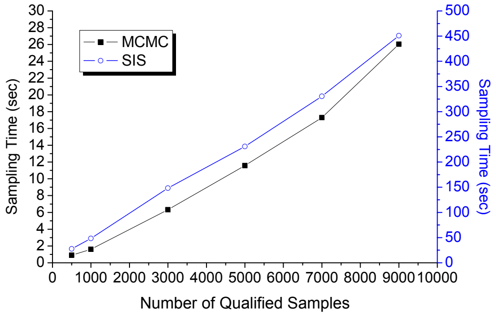


Figure 11: MCMC versus SIS on sampling time.

6.2.2 The Reconstruction Accuracy

In this experiment, we varied the number of qualified samples, data redundancy degree, and the number of managed racks per reader to investigate their effects on the reconstruction accuracy.

The Impact of the Number of Qualified Samples

We first increased the number of qualified samples from 500 to 9000 to investigate the performance of MCMC and SIS on reconstruction accuracy. Here we assumed that the read rate in the major detection region is 95% and the number of racks managed by a reader is 1. As demonstrated in Figure 12(a), with the increase of the number of qualified samples, the K-L divergence values of both approaches kept decreasing. However, MCMC always outperformed SIS with all experimented sample numbers. Particularly, when we drew 500 qualified samples, the K-L divergence of MCMC was 0.86 while the K-L divergence of SIS was 3.78. When we picked 9000 qualified samples, the K-L divergence of MCMC reduced to a remarkable value 0.64 comparing with 2.77 of the SIS solution. Also, as far as the top-1 success rate is concerned, with the increase of the number of qualified samples, the top-1 success rate of MCMC increased from 0.50 to 0.70 while the top-1 success rate of SIS extended from 0.36 to 0.55 as shown in Figure 12(b). In addition, the top-2 success rate of MCMC raised from 0.70 to 0.89 while the top-2 success rate of SIS changed from 0.60 to 0.76 as demonstrated in Figure 12(c).

The Impact of the Redundancy Degree

Next, we studied the performance of MCMC and SIS on the reconstruction accuracy by varying the data redundancy degree. Because the false positives are actually the successful readings about the objects in the minor detection regions of readers, we use the read rate in the minor detection region to define the data redundancy degree. The larger redundancy degree indicates the higher probability that a reader can detect an object in the neighboring zones (or racks).

Definition The *data redundancy degree* is the probability that a reader successfully detects a object in the minor detection region of that reader.

Here, we varied the data redundancy degree from 0.325 to 0.475, corresponding to the read rate in the major detection regions from

65% (the least reliable reader) to 95% (the most reliable reader). For each experiment, we drew 5000 qualified samples and the number of racks managed by a reader is 1. Figure 13 illustrates the results. With the enlargement of data redundancy degree, both the performances of MCMC and SIS on reconstruction accuracy are elevated. Specifically, as demonstrated in Figure 13(a), MCMC always maintained a lower K-L divergence value than SIS, reflecting a more precise prediction. Furthermore, as shown in Figure 13(b), the top-1 success rate of MCMC increased from 0.54 to 0.65 with the increase of the data redundancy degree while the top-1 success rate of SIS expanded from 0.42 to 0.51. Figure 13(c) demonstrates how the top-2 success rate increased when we raised the redundancy degree for MCMC and SIS.

The Impact of the Number of Managed Racks per Reader

We evaluated the performance of MCMC and SIS by varying the number of managed racks per reader. In order to deploy readers in a warehouse more efficiently, users may want to assign multiple racks to be managed by a single reader. Taking into account the fact that the overall detection region of a regular RFID reader has little chance to be more than 20 feet [17], we changed the number of racks managed by a reader from 1 to 6. As Figure 14(a) demonstrates, when each rack had its own reader, the K-L divergence values of MCMC and SIS were 0.68 and 3.11, respectively. When a reader monitored more racks, both the K-L divergence values of MCMC and SIS deteriorated. When a reader was responsible for detecting cases on six racks, the K-L divergence values raised to 1.66 of MCMC and 4.01 of SIS. Moreover, as demonstrated in Figure 14(b), with the enlargement of the number of managed racks per reader the top-1 success rate of MCMC decreased from 0.65 to 0.55. On the other hand, the top-1 success rate of SIS dropped from 0.51 to 0.41. Also, Figure 14(c) depicts how the top-2 success rate of MCMC and SIS decreased correspondingly.

6.3 Query Results

In our simulations, we focus on two types of queries: *location queries* and *remaining capacity queries*. Given an object and a zone, a *location query* returns the probability that the object is in that zone. On the other hand, given a zone, a *remaining capacity query* returns the leftover volume of a certain resource available in that zone. To show results of the above queries, for the simplicity of presentation, we issued queries against the small-scale warehouse setting. As Table 5 shows, we assume that there are twenty cases with identical length and five racks with a length limit to accommodate at most seven cases. True distributions of the first six cases are shown in Table 6(a) where the rows represent cases and columns represent racks. To be specific, if a case is on a rack, the corresponding position is 1 and 0 otherwise. Therefore, according to Table 6(a), the first case is on the first rack, the second case is on the fourth rack and so on. On the other hand, the corresponding noisy RFID data matrix, generated by our simulator, is shown in Table 6(b) in the same format. By comparing the two matrices, we can easily see that there is notable difference (existence of noise and duplicate readings) between these two matrices. Specifically, duplicate readings of the fourth and fifth case (consecutive 1's) occur in the noisy raw data. Afterward, we employed MCMC and SIS to cleanse the noisy distribution matrix to recover the true distributions. In this experiment, we drew 5000 qualified samples for MCMC and SIS, respectively, and we assume that the read rate in the major detection region of readers is 95%.

$$\underset{(a)}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}} \quad \underset{(b)}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}}$$

Table 6: The matrix of the first six cases: (a) true distribution and (b) noisy raw data.

6.3.1 *Location Queries*

The results of the location queries returned by MCMC and SIS for the first case and fourth case are demonstrated in Figure 15(a) and Figure 15(b), respectively. For the first case, the correct location is the first rack. MCMC predicted a probability of 0.47 on that rack while SIS predicted a probability of only 0.01. On the other hand, for the fourth case, the exact location is the second rack. MCMC predicted a probability of 0.51 on that rack while SIS predicted a probability of 0.32. In summary, MCMC tends to generate a more precise probability distribution than SIS does. In other words, MCMC provides a superior overall prediction of the distribution of all the objects monitored by an RFID system compared to SIS.

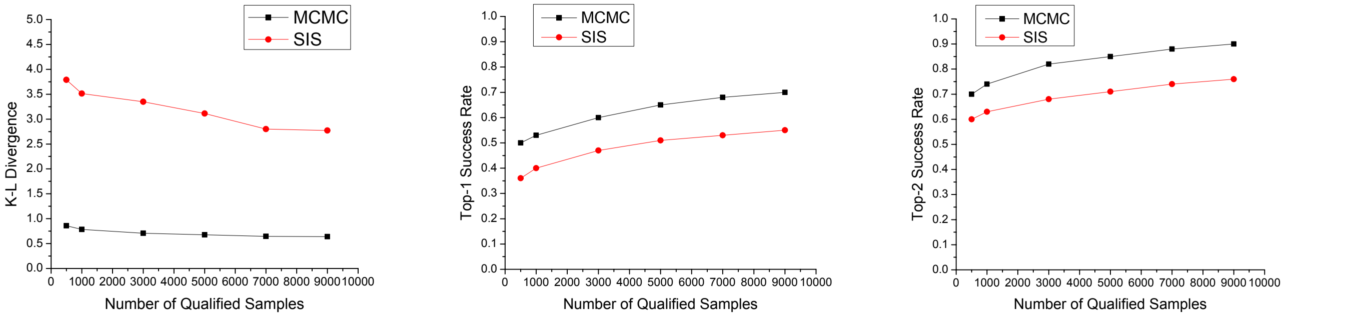
6.3.2 *Remaining Capacity Queries*

We applied the available length on a rack as the acquirable volume of that rack to demonstrate the remaining capacity query. The results of the remaining capacity queries answered by MCMC for the racks 1, 2 and 3 are demonstrated in Figure 15(c). Note that the correct remaining capacity on each rack is 3 because each rack exactly accommodates 4 cases according to the true distribution matrix used for this experiment (the first six rows of the true distribution matrix are as shown in Table 6(a)). As illustrated in Figure 15(c), the remaining length on racks 1 was reported correctly. To be specific, MCMC reported that the available length on rack 1 was 3 with a probability of 81%. On the contrary, MCMC predicted the available length on rack 2 as 1 with a probability of 52% and estimated the available length on rack 3 as 2 with a probability of 46% (i.e., the remaining capacities of both rack 2 and 3 were slightly underestimated).

7. RELATED WORK

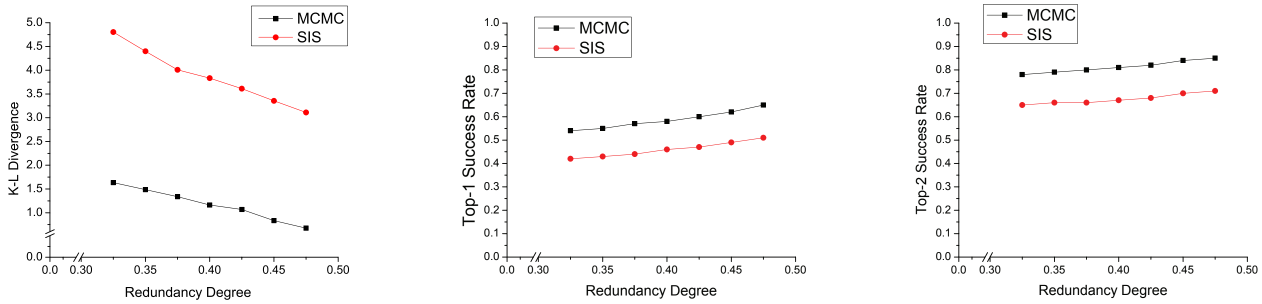
Many systems have been developed to manage data with uncertainty. The usual approach to address uncertainty is to augment the classical relational model with attribute-level or tuple-level probability values [1, 3, 4, 6, 7]. On the other hand, a more general approach based on sampling is proposed for managing incomplete and uncertain data [14, 28]. The idea is simple and intuitive: we construct random samples while observing the prior statistical knowledge and constraints about the data. Thus, each sample is one possible realization in the space of uncertainty, and the entire set of samples reveals the distribution of the uncertain data we want to model. Queries and inferences are then conducted against this distribution.

An important application that drives the recent surge of interest in managing incomplete and uncertain data is RFID data management. Chawathe et al. [5] proposed a system architecture of a distributed RFID system and discussed related data management challenges such as inferences and online warehousing. An expressive temporal data model for RFID data is defined by Wang and Liu [26] to support tracking and monitoring queries. Gonzalez et al. [12] designed a warehousing model which provides RFID data compression and path-dependent aggregates. Based on the



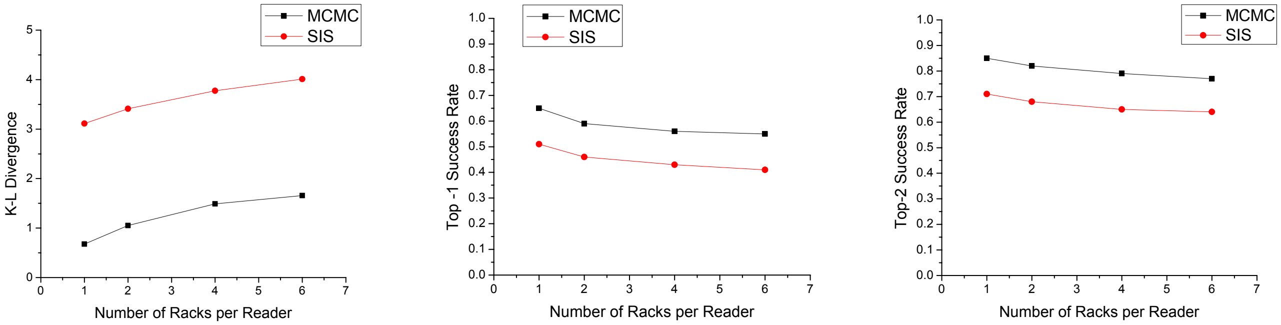
(a) K-L divergence of 5000 cases. (b) Top-1 success rate of 5000 cases. (c) Top-2 success rate of 5000 cases.

Figure 12: The impact of the number of qualified samples.



(a) K-L divergence of 5000 cases. (b) Top-1 success rate of 5000 cases. (c) Top-2 success rate of 5000 cases.

Figure 13: The impact of RFID redundancy degree.

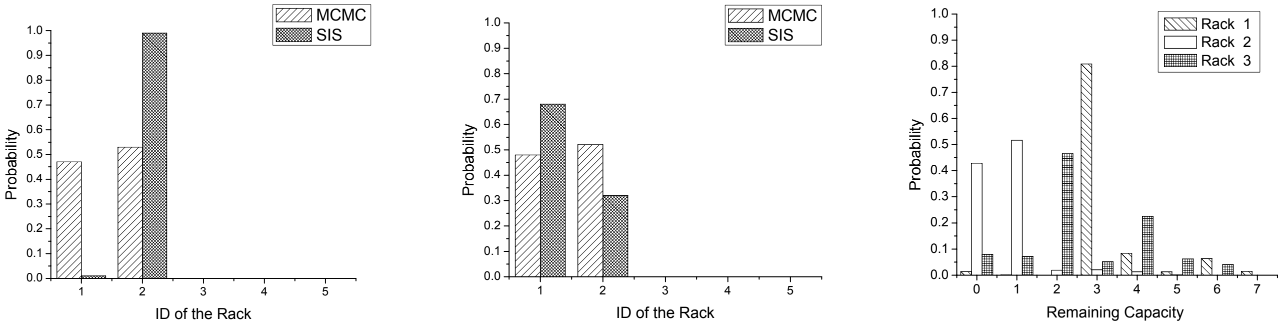


(a) K-L divergence of 5000 cases. (b) Top-1 success rate of 5000 cases. (c) Top-2 success rate of 5000 cases.

Figure 14: The impact of the number of managed racks per reader.

proposed warehousing framework, they also developed techniques for summarizing and indexing RFID data and various query methods. Because RFID raw data usually contains anomalies [10], solutions have been proposed to clean the input data from readers. Jeffery et al. [15] presented a framework for building data cleaning infrastructure to support pervasive applications. An adaptive smoothing filter (SMURF) for RFID data cleaning was proposed in [17]. SMURF focuses on a sliding-window aggregate that interpolates for lost readings. SMURF models the unreliability of RFID readings by taking RFID streams as a statistical sample of physical tags, and exploits techniques in sampling theory to drive its cleaning processes. Considering different anomaly definitions between applications, Rao et al. [21] introduced a deferred approach for detecting and correcting RFID data anomalies by utilizing declarative sequenced-based rules. Their approach allows application specific cleansing at query time. Our paper, however, is motivated by how to leverage data redundancy to elevate the localization accuracy for all the tagged objects in a target area.

The works in [18, 19, 28] are the most relevant research to this paper. For correcting erroneous RFID raw data, Khoussainova et al. [18, 19] presented a system for correcting input data errors automatically using application defined global integrity constraints. The system corrects the input data by inserting missing tuples when necessary and assigning to each one the probability that it is correct for groups of conflicting tuples. This maximum entropy based solution is practical. However, it is unable to capture all application related prior knowledge and dependency compared with sampling-based approaches. Useful information can be recovered from noisy RFID data by exploiting constraints and sequential importance sampling methods [28]. Nevertheless, the work in [28] failed to consider the duplicate readings caused by the overlapped detection regions of RFID readers.



(a) Location query for the first case (true location: rack 1).

(b) Location query for the fourth case (true location: rack 2).

(c) Remaining capacity queries for racks 1, 2 and 3 (true available length: 3).

Figure 15: Example query results answered by MCMC and SIS.

8. CONCLUSIONS

The data reported by RFID devices are known to be unreliable. In this research, we propose a Bayesian inference based approach for cleansing RFID raw data which can take advantage of duplicate readings. In order to evaluate the location and aggregate queries, our approach employs prior knowledge to quantify the degree of uncertainty on the location of each object and the remaining capacity in each zone. Furthermore, we propose the n-state model to capture likelihood and validate that the 3-state model can maximize the system performance. Finally, we devise MH-C to efficiently sample from the posterior distribution under environmental constraints.

9. REFERENCES

[1] P. Agrawal, O. Benjelloun, A. D. Sarma, C. Hayworth, S. Nabar, T. Sugihara, and J. Widom. Trio: A System for Data, Uncertainty, and Lineage. In *VLDB*, pages 1151–1154, 2006.

[2] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An Introduction to MCMC for Machine Learning. *Machine Learning*, 50(1-2):5–43, 2003.

[3] P. Andritsos, A. Fuxman, and R. J. Miller. Clean Answers over Dirty Databases: A Probabilistic Approach. In *ICDE*, page 30, 2006.

[4] L. Antova, C. Koch, and D. Olteanu. Query Language Support for Incomplete Information in the MayBMS System. In *VLDB*, pages 1422–1425, 2007.

[5] S. S. Chawathe, V. Krishnamurthy, S. Ramachandran, and S. E. Sarma. Managing RFID Data. In *VLDB*, pages 1189–1195, 2004.

[6] R. Cheng, S. Singh, and S. Prabhakar. U-DBMS: A Database System for Managing Constantly-evolving Data. In *VLDB*, pages 1271–1274, 2005.

[7] N. Dalvi and D. Suciu. Efficient Query Evaluation on Probabilistic Databases. *The VLDB Journal*, 16(4):523–544, 2007.

[8] A. Deshpande, C. Guestrin, and S. Madden. Using probabilistic models for data management in acquisitional environments. In *CIDR*, pages 317–328, 2005.

[9] D. W. Engels and S. E. Sarma. The Reader Collision Problem. In *IEEE SMC*, 2002.

[10] C. Floerkemeier and M. Lampe. Issues with RFID Usage in Ubiquitous Computing Applications. In *Pervasive*, pages 188–193, 2004.

[11] M. J. Franklin, S. R. Jeffery, S. Krishnamurthy, F. Reiss, S. Rizvi, E. Wu, O. Cooper, A. Edakkunni, and W. Hong. Design Considerations for High Fan-In Systems: The HiFi Approach. In *CIDR*, pages 290–304, 2005.

[12] H. Gonzalez, J. Han, X. Li, and D. Klabjan. Warehousing and Analyzing Massive RFID Data Sets. In *ICDE*, page 83, 2006.

[13] J. Ho, D. W. Engels, and S. E. Sarma. HiQ: A Hierarchical Q-learning Algorithm to Solve the Reader Collision Problem. In *SAINT Workshops*, pages 88–91, 2006.

[14] R. Jampani, F. Xu, M. Wu, L. L. Perez, C. Jermaine, and P. J. Haas. MCDB: A Monte Carlo Approach to Managing Uncertain Data. In *SIGMOD*, pages 687–700, 2008.

[15] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom. Declarative Support for Sensor Data Cleaning. In *Pervasive*, pages 83–100, 2006.

[16] S. R. Jeffery, M. J. Franklin, and M. N. Garofalakis. An Adaptive RFID Middleware for Supporting Metaphysical Data Independence. *VLDB J.*, 17(2):265–289, 2008.

[17] S. R. Jeffery, M. N. Garofalakis, and M. J. Franklin. Adaptive Cleaning for RFID Data Streams. In *VLDB*, pages 163–174, 2006.

[18] N. Khoussainova, M. Balazinska, and D. Suciu. Towards Correcting Input Data Errors Probabilistically Using Integrity Constraints. In *MobiDE*, pages 43–50, 2006.

[19] N. Khoussainova, M. Balazinska, and D. Suciu. Probabilistic Event Extraction from RFID Data. In *ICDE*, pages 1480–1482, 2008.

[20] J. Myung, W. Lee, J. Srivastava, and T. K. Shih. Tag-Splitting: Adaptive Collision Arbitration Protocols for RFID Tag Identification. *IEEE Trans. Parallel Distrib. Syst.*, 18(6):763–775, 2007.

[21] J. Rao, S. Doraiswamy, H. Thakkar, and L. S. Colby. A Deferred Cleansing Method for RFID Data Analytics. In *VLDB*, pages 175–186, 2006.

[22] S. M. Ross. *Introduction to Probability Models, Ninth Edition*. Academic Press, 2006.

[23] L. Sullivan. RFID Implementation Challenges Persist, All This Time Later. *InformationWeek*, October 2005.

[24] T. Tran, C. Sutton, R. Cocci, Y. Nie, Y. Diao, and P. Shenoy. Probabilistic Inference over RFID Streams in Mobile Environments. In *ICDE*, 2009.

[25] J. Waldrop, D. W. Engels, and S. E. Sarma. Colorwave: An Anticollision Algorithm for the Reader Collision Problem. In *ICC*, pages 1206– 1210, 2003.

[26] F. Wang and P. Liu. Temporal Management of RFID Data. In *VLDB*, pages 1128–1139, 2005.

[27] R. Want. The Magic of RFID. *ACM Queue*, 2(7):40–48, 2004.

[28] J. Xie, J. Yang, Y. Chen, H. Wang, and P. S. Yu. A Sampling-Based Approach to Information Recovery. In *ICDE*, pages 476–485, 2008.