Measuring the Uncertainty of RFID Data Streams

Yaozong LIU¹,²,*, Hong ZHANG¹, Lingling ZHANG², Xuesong XU³

¹School of Computer and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China
²Department of Information Technology, Nanjing Forest Police College, Nanjing 210023, China
³College of Information and Technology, Nanjing University of Chinese Medicine, Nanjing 210046, China

Abstract

RFID technology is widely used in the fields of location and position awareness. The RFID data streams are uncertain and need accurate and timely processing, since they are influenced by many factors. We propose an uncertain data optimal algorithm for RFID data streams. By analyzing the existing deficiency in the conventional particle filter method, we use a method based on entropy to derive the optimal weighting of features. We also use possible degree matrix to choose the best particle, capture the current state of the object effectively from uncertain RFID data streams. The optimization results of algorithms make sampling set move to the region that has a larger posteriori density value, so as to increase the calculation efficiency and significantly reduce the required number of particles for precise positioning. Finally, the experiments show that this method can effectively measure the uncertainty contained in the original RFID data streams.

Keywords: Uncertainty; RFID; Data Streams; Optimal Estimation; Particle Filter

1 Introduction

At present, Radio Frequency Identification (RFID) is applied in the location and position awareness for moving objects [1]. The positioning technology based on RFID needs the mobile objects positioning data recorded by the reader, but does not require the communication among the moving objects which carry RFID tags. The moving objects which carry tags continuously send their position and features to the reader. These location streams collected by RFID tags are influenced by the sensor itself and the scattering and collision of signal in the surrounding environment. Therefore the tag data received by the RFID reader is uncertain, which causes a lot of difficulties for RFID location and position-aware applications. How to effectively deal with the uncertainty of RFID data so as to improve the positioning precision for the moving objects, and the real-time performance is essential in RFID studies.
In the RFID application, the relationships that exist in the RFID read process, lead to many uncertainties in the results of the query of some attributes, such as temperature and humidity. Since there exist multiple possible instances, for these probabilistic databases, different query plans may return different probability values as the query results. The basic reason is that the correlation between the probability of data in the process of the design of a query plan is not being considered, leading to repeated calculation [2]. To solve this problem, the additional possible probabilities must be attached to the RFID data before their first inflow into the processing system, namely, uncertainty of RFID data need to be measured.

At present, there are only very limited studies on the uncertainty measurement of RFID data. The existing probability estimation method cannot be directly applied to the RFID data management. The use of lineage to simulate the uncertainty and the query results of the data was firstly discussed in [3]. How to improve the efficiency of uncertain data processing has been studied in [4], by using the probability from the result of the global Boolean formula on tuples. However, the existing works all assumed that the original data probability is known (that is, the preset great uncertainty), while not mentioned the probability measurement of original data, especially the measurement for the probability of the multi-objective sensor data. Various methods for optimizing the quality of resampling particle have been proposed in [5-7]. The number of particles is adjusted adaptively according to the situation of RFID data evolve, to improve the particle degradation phenomenon, at the same time by using the improved Particle Swarm Optimization (PSO) to optimize the resampling performance, improve the problem of particle impoverishment, and provide the uncertainty measurement contained in original data source. But the core of particle filter algorithm is to represent the posterior probability density by using the weight of a series of random samples. The particle which has the maximum weight represents the system state with the maximum likelihood. This requires an additional number of particles, while as the particle number increases, after several iterations, it is difficult to converge to the real state; computational efficiency is greatly reduced at the same time, which cannot meet the requirement of the real time RFID data stream. To this end, we propose an uncertain data optimal algorithm for complete data streams. This algorithm uses the dependence of tuple on the attributes to process the weight integration. By choosing the particles having the optimal weights, we can effectively capture the current state of the object from the uncertain RFID data stream, reduce the required number of particles at the same time, adapt to one scan RFID data algorithm, and satisfy the requirement of real-time processing.

2 RFID Data Streams Models and Definitions

Definition 2.1 Set $S = (X, A)$ is a data stream, $X$ is the non-empty finite set of objects, $A$ is the non-empty finite set of attributes, any attribute $a \in A$, $a: X \rightarrow V_a$ is a map, in which the $V_a$ is referred to as a value set of $a$. For a given data stream $S = (X, A)$, if any attributes $a \in A$, and $V_a$ does not contain null values, $S$ is a complete data stream.

Definition 2.2 RFID data stream $S$ is composed of multiple complete data streams $S'$, the $S'$ part in the sliding window is equal to the tables in the relational data model, the tuple $x \in S$ is composed of multiple attribute values.

Let $p_{avg}^i$ refers to observation probability for a RFID tag $i$ in a window, if the number of the tags in the window $n_i$ satisfies the inequality $n_i \geq \left\lceil \frac{\ln(1/p)}{p_{avg}^i} \right\rceil$, then it is guaranteed that the tag $i$ can
be read in the window \( n_t \) with a probability greater than \([8]\).

**Definition 2.3** The particle filter based on the measure values and the control values, use a weight point set \( \{ (x^i, \tilde{w}^i), i = 1, 2, \cdots, N \} \) to approximate the posterior probability distribution. According to the application background of RFID data stream, we use the following nonlinear model with multiply noise \([9]\):

\[
\begin{align*}
  x_t &= f(x_{t-1})(1 + v_{t-1}) \\
  y_t &= h(x_t)(1 + u_t)
\end{align*}
\]

Where the \( y_t \) is the measure values,

\[
\begin{align*}
  f(x_t) &= 0.5x_t + \frac{25x_t}{1 + x_t^2} + 8 \cos(1.2t) \\
  h(x_t) &= \frac{x_t^2}{20};
\end{align*}
\]

\( v_t \) and \( u_t \) are the white noise with the mean at 0, the variances \( Q \) and \( R \).

3 **Attribute Weights Optimization Models and Definitions**

The data uncertainty of RFID complete data stream can be subdivided into tuple level uncertainty and attribute level uncertainty. Tuple level uncertainty describes the presence or absence of a tuple, and attribute level uncertainty does not involve the entire tuples uncertainty. Since the query results are returned in the form of data stream, to produce the query results with same data but different probability values, which lead to repeated calculation. Therefore, the dependence that uncertainty tuples on attributes must be considered, so as to increase the corresponding weights for attributes, get set-valued optimization matrix for each tuple, and then process the optimal selection for tuples according to the set-valued optimization matrix.

**Definition 3.1** There are two uncertain variables \( b_1 = [b_{a1} \ b_{b1}] \) and \( b_2 = [b_{a2} \ b_{b2}] \), where \( b_a \) and \( b_b \) are lower limit and upper limit respectively, \( \text{len}(b_1) = \beta_1 \alpha_1 \) and \( \text{len}(b_2) = \beta_2 \alpha_2 \) represent the length of the two uncertain variables; then we say

\[
P(b_1 \geq b_2) = \frac{\max(0, \text{len}(b_1) + \text{len}(b_2) - \max(\beta_2 - \alpha_1, 0))}{\text{len}(b_1) + \text{len}(b_2)}
\]

**Definition 3.2** The distance between \( b_1 \) and \( b_2 \) is:

\[
d(b_1, b_2) = 1/2(\beta_1 - \beta_2 + \alpha_1 - \alpha_2)
\]

**Definition 3.3** The uncertain tuples \( \{x_1, x_2, \cdots, x_m\} \) which have same data but different sampling probability values in every \( \Delta t \) seconds from the sliding window, the attributes set of tuples is \( A = \{a_1, a_2, \cdots, a_n\} \). For a tuple \( x \in X \), measure according to the attribute \( a_j \), and get uncertain data stream weight optimization matrix \( R = (r_{ij})_{m \times n} \), in which \( w = (w_1, w_3, w_3, \cdots, w_n)^T \) is attribute weights vector, where \( w_j \in (0, 1), \sum_i w_j = 1 \).

At the same time, for the weight optimization matrix, we choose the entropy as a measurement for uncertainty. Smaller entropy means more certainty the variables have, namely when the
entropy is very small, we can treat this variable as it is certain, and treat the value which has the 
highest return probability as the uncertain variable values. The \( w \) uncertain correction values 
which have \( n \) different possible value can be represented by the entropy:

\[
H(w) = -\sum_{i=1}^{n} w_i T(w_i) \log_{e} T(w_i)
\]  

(5)

The \( T(b_i) \) is probability quality function for \( B \) valued \( b_i \). According to the definition of entropy, 
the number of possible value that entropy to variables \( B \) is sensitive. For example, the entropy 
for uncertain variables with 10 identical probability values will be greater than the entropy for 
uncertain variables with 3 identical probability values. To reduce this effect, the maximum entropy 
with \( n \) possible values is used, to process the normalization for uncertain variables with \( n \) possible 
values:

\[
H(w) = (\frac{-\sum_{i=1}^{n} w_i T(w_i)}{n} \log_{e} (\frac{1}{n}))
\]

(6)

**Theorem 1** There must be optimal tuples for complementary judgment matrix \( P \).

**Proof:** By adjusting the size of the deviation variables, the weight that can meet comple-
mentary judgment matrix \( P \) related constraint will be able to get, that is the feasible region 
of the complementary judgment matrix \( P \) is not empty, so there must be optimal tuples for complementary judgment matrix \( P \).

QED.

4 Algorithm Description

In order to satisfy the online measurement of RFID data streams uncertainty, in this paper, we 
improve the particle filter algorithm, and propose a novel uncertain data optimal algorithm for 
RFID data streams. The algorithm is named optimal estimation particle filter (OEPF), and 
described as follows:

**Step 1** Obtain measured values. Use the fitness function from literature [5] to initialize par-
ticles, design the appropriate sliding window size. The input is tag average probability and 
confidence probability. The output is the appropriate window size \( F \).

**Step 2** At time \( \Delta t \), sample \( L \) particles from the importance density function, represented as 
\( \{x^i_k, w^i_k\}_{i=1}^{L} \), set the initial weight for each sample to \( w^i_k = 1/L, i = 1, 2, \cdots, L \), the importance 
density function chooses the transfer prior probability \( x^i_k = q(x^i_k|x_{k-1}^{i}, y_k) = p(x^i_k|x_{k-1}^{i}) \).

**Step 3** Update the particle weights according to the latest measured value:

\[
\hat{w} = \hat{w}_{k-1} p(y_k|x_{k-1}^{i}) = \hat{w}_{k-1} \frac{p(y_k|x_{k}^{i})p(x_{k}^{i}|x_{k-1}^{i})}{q(x_{k}^{i}|x_{k-1}^{i}, y_k)}
\]

(7)

**Step 4** Update each particle’s speed and position by using the PSO algorithm from literature 
[7], to make the particles close to the real state. Set threshold \( \varepsilon \), then determine whether the variance is less than threshold \( \varepsilon \), if the variance is not less than the threshold, then go to Step 6, 
otherwise continue.
Step 5 By comparing the individual extreme value of particles, use PSO algorithm to update the particle velocity and position, get rid of the subprime position, jump out of local optimum, and drive the particles to be near global optimal location.

Step 6 Particle weight normalization: \( \hat{w}_k^i = \hat{w}_k^i / \sum_{i=1}^{L} \hat{w}_k^i \).

Step 7 Resampling: when \( L_{eff} = \frac{1}{\sum_{i=1}^{L} (\hat{w}_k^i)^2} < L_{\text{threshold}} \), resampling the original particle \( \{x_k^i, \hat{w}_k^i\}_{i=1}^{L} \), and get particles \( \{x_k^i, L^{-1}\}_{i=1}^{L} \) equal in weighted.

Step 8 Select \( m \) particles which have same data but different probability values, according to the \( n \) properties, get the uncertain weight optimized matrix \( R \).

Step 9 Use normalization processing in Eq. (6), calculate the optimal weight vector \( w_j \) of attributes, and obtain the comprehensive attribute value \( z_i = \sum_{j=1}^{n} r_{ij} w_j \) for each particle.

Step 10 Process the pairs comparison for uncertain variables by using the possible degrees from Eq. (4), and construct complementary judgment matrix \( P \).

Step 11 According to the nature of the complementary judgment matrix \( P \), construct a simple formula:

\[
\hat{w} = \frac{1}{m(m-1)} \left[ \sum_{j=1}^{m} p_{kj} + \frac{m}{2} - 1 \right], \ i \in M
\]

Get the sequencing vector \( w = \{w_1, w_2, \ldots, w_n\}^T \) in the matrix \( P \), then sort and merit by the particle size.

Step 12 Particle state estimation: \( \hat{x} = \sum_{i=1}^{L} \hat{w}_k^i x_k^i \); particles variance estimation: \( p_k = \sum_{i=1}^{L} \hat{w}_k^i (x_k^i - \hat{x}_i) (x_k^i - \hat{x})^T \).

Step 13 If the \( t \) moment is the last moment for object, then end the algorithm. Otherwise, set \( t = t + 1 \), and return to Step 1, recursive estimate posterior probability for object state at next time \( \Delta t \).

The essence of PSO algorithm is using own information, individual extreme value and global extreme value information to guide the particle’s position in the next iteration. After optimization, the particles set trend more toward to high likelihood area before the weights update, that solve the problem of particle impoverishment. At the same time, the particle number is adjusted adaptively to improve the particle degradation phenomenon. Step 8 to 11 perform sampling in the sliding window, choose the best tuple by using the complementary judgment matrix, to further reduce the number of particles, so as to improve the efficiency of the algorithm to satisfy the requirement of real time RFID data stream.

5 Experiments

OEPS proposed in this paper can be applied to locate and track moving people or objects in electromagnetic interference-free office environment. We randomly set 20 RFID Tags distribution in a 8M*8M square in the laboratory, treat the RFID data samples that reflect the location information as particles. Students were asked to carry tags and make random and uniform movement in the recognition scope of 20 readers, experiment in three different environments. The sampling interval is 0.4 s, the reading rate of reader is 0.5-1, and the computer system is: CPU: Intel Core TM 2 Duo (2.9 GHz)/main memory to 4 GB. Each RFID sample is a particle, and its attributes include sensory information such as height, speed and position. The models
from Eq. (1) and Eq. (2) were used to test our algorithm. The sliding window uses 60 time steps \( \Delta t = 1, 2, \cdots, 60 \). In order to verify the validity of the algorithm, we compare the PSOPF algorithm from literature [7].

With the white noise whose variances are \( Q \) and \( R \), given in the definition of Eqs. (1) and (2), the filtering performance were compared between two algorithms. As shown in Table 1 and Table 2, in the same noise environment, our algorithm has maximum number of valid samples, higher efficiency in increasing the diversity and suppressing degradation of particles compare to the PSOPF algorithm. The estimation accuracy of PSOPF algorithm is less than our algorithm even the number of particles is increased to 800, while the time cost of PSOPF algorithm estimation is higher, indicating that under the same accuracy requirements, the proposed algorithm improves the efficiency of the algorithm. Meanwhile, in the case of an increase of noise, our algorithm has the best anti-noise performance, restrain the particle degradation even in the case of an increase noise, increase particle diversity, and maintain the accuracy of algorithm estimation.

### Table 1: Comparison filtering performance of two algorithms with \( R=10 \) and \( Q=1 \)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Number of particles</th>
<th>Average number of valid samples</th>
<th>Mean square error</th>
<th>Running time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSOPF</td>
<td>400</td>
<td>48</td>
<td>2.63</td>
<td>1.05</td>
</tr>
<tr>
<td>PSOPF</td>
<td>800</td>
<td>86</td>
<td>2.46</td>
<td>3.32</td>
</tr>
<tr>
<td>OEPF</td>
<td>400</td>
<td>98</td>
<td>1.69</td>
<td>2.36</td>
</tr>
<tr>
<td>OEPF</td>
<td>800</td>
<td>182</td>
<td>1.21</td>
<td>3.32</td>
</tr>
</tbody>
</table>

### Table 2: Comparison filtering performance of two algorithms with \( R=20 \) and \( Q=1 \)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Number of particles</th>
<th>Average number of valid samples</th>
<th>Mean square error</th>
<th>Running time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSOPF</td>
<td>400</td>
<td>52</td>
<td>4.56</td>
<td>1.34</td>
</tr>
<tr>
<td>PSOPF</td>
<td>800</td>
<td>94</td>
<td>4.15</td>
<td>3.56</td>
</tr>
<tr>
<td>OEPF</td>
<td>400</td>
<td>106</td>
<td>2.86</td>
<td>2.62</td>
</tr>
<tr>
<td>OEPF</td>
<td>800</td>
<td>194</td>
<td>1.33</td>
<td>3.55</td>
</tr>
</tbody>
</table>

### 6 Conclusions

In recent years, as an emerging filtering algorithm, the particle filter has gained a lot of attention. While it is required the particle with the maximum weight value indicates the state that the system is most likely to be, it would increase the number of particles. Taking the features of RFID data streams into account, there are still some weakness need to be solved in practice. In this paper, we derived the attention optimal weights by using entropy method, and choose the best particles by using the possible degree matrix. Compared with existing algorithms, our algorithm
can get better confirm information for the particles, meanwhile, exclude particles which have low weight values and improve the algorithm efficiency. Experiment results showed that our method can result in more accurate data, has good efficiency, and is very suitable for RFID moving object location estimation.

Acknowledgements

This work is supported by the Fundamental Research Funds for the Central Universities (L-GZD201402, No. LGYB201412).

References


