TELKOMNIKA, Vol.11, No.3, March 2013, pp. 1707~ 1713
ISSN: 2302-4046

Efficient RFID Data Cleaning Method

Li Xing, Fu Wen-Xiu
School of Electronic and Information Engineering, Beijing Jiaotong University.
Beijing, 100044, China
e-mail: lixingbjtu@163.com

Abstract

RFID (Radio Frequency Identification) technology transfers data between movable tagged objects and readers without line of sight, and the captured data tends to be noisy. The inherent unreliability makes the data unreliable to application. Nowadays, the main solution is to use sliding window, but it is difficult to decide the window size, especially when the tag moves frequently or with high false positive. To solve the mentioned problems, SWKF (Sliding Window based on Kalman Filter Pre-processing) is proposed. It preprocesses the RFID data to make the read rate close to the real one, detects and filters the mobile tags. Then, the preprocessed data is smoothed to further improve accuracy. At the same time, the mid-window slide point reduces the output. Through the combination of Kalman Filter and sliding window, SWKF provides accurate RFID data to application.

Keywords: *radio frequency identification, Kalman filter, preprocess, sliding window*

1. Introduction

Combined with Internet and telecommunication, RFID technology can achieve global scale items tracking and information sharing [1,2]. With the increasing number of RFID data and the inherent unreliability [3], it is necessary to clean the collected data to satisfy RFID application [4-8]. The errors occurring in the process of data capture often include false negative and false positive. The observed read rate in real-world RFID deployments is often in the 60%-70% range [3,9], that is to say, over 30% of the tag readings are routinely dropped.

Many scholars have worked on the problems mentioned above. The method based on sliding window [4,5,10,11] is the typical approach, but it is hard to decide the window size. With lots of interference, especially tags in mobile condition that is hard to be detected, it increases the error data. Methods proposed in [7,12] can correct the read rate dynamically, but they produce large amounts of output data.

In this paper, SWKF is presented. We summarize our contributions as follows:

• The combination of Kalman Filter and sliding window. The Kalman Filter pre-processing corrects the read rate to some extent, removes the interference, detects and filters mobile tags.

• Smoothing the pre-processed data. It avoids the window jittering phenomenon. This improves the effect of the sliding window processing. The mid-window slide point decreases the storage space.

• Experiments. They show that the accuracy and effectiveness of the proposed algorithm.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 presents the structure of RFID data. Section 4 presents a detailed treatment of the cleaning method. Section 5 reports experimental and performance results. We conclude our study in Section 6.

2. Related Work

For RIFD data cleaning, Smooth is first proposed in the EPC Global Reader Protocol [10], the original purpose is to represent a large number of tag stream events with the logical meaningful events. But it actually plays the rule of smoothing event stream to clean the false negative. Its window size is fixed.

Jeffery proposes a statistical smoothing algorithm SMURF [5], which models the unreliability of RFID readings by viewing RFID streams as a statistical sample of tags in the physical world. It adapts the window size to provide accurate RFID data to application. But if the tags move rapidly, it can't address the window size. To solve this problem, Lingyong Meng et al.

Received June 17, 2012; Revised September 10, 2012; Accepted October 19, 2012

[11] proposes a new and improved algorithm, which considers factor such as reader communication range, velocity of tag movement, and reading frequency in deterring the size of window.

Jeffrey *et al.* [4] presents ESP, a declarative query-based framework. ESP consists of a programmable pipeline of declarative query-based stages. These stages segment the cleaning process into five tasks. These five approaches have increasing levels of functionality. ESP uses windowed processing to group reading within a granule, cleans data based on temporal and spatial correction, but it isn't related to how to set it. It mainly solves false negative and false positive.

Gonzalez *et al.* [7] presents a new cleaning method based on Dynamic Bayesian Networks (DBNs). It corrects the tag read rate dynamically, and considers the observations and estimates, but they are obtained by the historical data, and can't be updated dynamically.

When the missed read rate is high, especially when tags move rapidly, the effect of the mentioned algorithms above is bad.

Wang Yan *et al.* [12] proposes a cleaning method that used Kalman Filter, which solves false negative and false positive from single readers, but needs much space to store the tags.

PSCleaning based on pseudo event is proposed in [13]. It reduces the time delay of data output by introducing the notion of pseudo event into sliding window, and decreases the volume of output by handling false positive and duplicate readings at the same time. But it doesn't improve the accuracy. There, we present SWKF, which combines Kalman Filter with sliding window.

3. RFID Data

RFID system generates a stream of data that results from interrogation cycles occurring at recurring time intervals at each Reader. The reading data generated in each interrogation cycle is usually a set of tuples of the form (EPC, Reader, time). The tuple can be added with extra information, such as the tag type (class 0, class 1, generation 2, etc), the antenna used by the Reader, or the power level of the interrogation signal. From the application perspective, it is necessary to look at multiple interrogation cycles as a single unit known as a read cycle. In such case, we get tuples of the form (EPC, Reader, time, responses), where responses is the number of interrogation cycles when the tag was read.

Definition 1 (epoch) That is, the read cycle mentioned above, a single unit. We assume that 10 interrogation cycles as a single unit.

Definition 2 (p_i) Read rate of the tag i in a epoch, $p_i = response/10$.

4. Cleaning Method

4.1. Kalman Filter Model

Kalman Filter consists of two parts: time update and measurement update. Time update process estimates the current state utilizing the optimal value of the time on a state. Measurement update process uses observations on the current status of amendments to update the estimates obtained from the previous time to get more accurate estimates, recycling to approximate the true value.

Linear differential equations of Kalman Filter:

Prediction equation: $X(k) = AX(k-1) + BU(k) + W(k)$

Observation equation: $Z(k) = HX(k) + V(k)$

$X(k)$ is the system state at time *k*, and $U(k)$ is the system control at time *k*. *A* and *B* are the system parameters (*A*=*B*=1). $Z(k)$ is the measured value at time *k*, *H* is the measurement system parameters (*H*=1). $W(k)$ and $V(k)$ denote the process and measurement noise. They are assumed to be Gaussian white noise, zero mean, and variance, respectively are *Q*, *R*.

First, we predict the read rate of RFID in the next cycle by utilizing the process model. The most widely used numerical prediction method is regression analysis. We utilize the most commonly used method in the regression analysis least squares fit.

Read rate for n cycles uses linear regression analysis of fitting a straight line y=a+bk, and b is as follows:

$$b = \frac{\sum_{i=1}^{n}(x_i - x)(y_i - y)}{\sum_{i=1}^{n}(x_i - x)^2} \quad (1)$$

$$X(k|k-1) = X(k-1|k-1) + b * \Delta t \quad (2)$$

In equation 2, $X(k|k-1)$ represents the optimal value at *k*-1, Δt represents a read cycle. Read rate predictions have been updated, but covariance of $X(k|k-1)$ is not updated. *P* denotes the covariance:

$$P(k|k-1) = P(k-1|k-1)A' + Q \quad (3)$$

In equation 3, $P(k|k-1)$ is the covariance of $X(k|k-1)$, $P(k-1|k-1)$ is the covariance of $X(k-1|k-1)$, A' is the transposed matrix of *A* (*A*=1), *Q* is the covariance of the system process. Equation 2, 3 forecast the system. With RFID read rate predictions, and then collect the observed rate of the RFID reader. Then, we can get the optimal estimate of current (k) RFID read rate from the prediction and measurement.

$$X(k|k) = X(k|k-1) + Kg(k)(Z(k) - X(k|k-1)) \quad (4)$$

Kg is the Kalman gain:

$$Kg(k) = P(k|k-1)/(P(k|k-1)H' + R) \quad (5)$$

Now, we has got the optimal RFID read rate $X(k|k)$ at time *k*. However, in order to make the Kalman Filter constantly running until the end of the system process, we need to update the covariance of $X(k|k)$ at *k*.

$$P(k|k) = (I - Kg(k)H)P(k|k-1) \quad (6)$$

When the system steps into *k*+1 state, $P(k|k)$ is the formula (3) $P(k-1|k-1)$. In this way, the algorithm could go autoregressive operator.

4.2. Sliding Window Model

It uses adapted sliding window, the key is the decision of the sliding window size. For each tag, it views each epoch as an independent Bernoulli trail $B(|W|, p_i)$, where |*W*| is the window size. If the tag *i* is read, and appears in *W*, then tag *i* meets: $i \in S$ and $S \in W$, where *W* denotes the set, p^{avg} denotes the estimation of p_i calculated based on all the epoch reports in the window, that is

$$p^{avg} = \frac{\sum_{i \in S}(p_i)}{|S|} \quad (7)$$

The value |S| follows Bernoulli distribution, $|S| \square B(|W|, p^{avg})$.

Based on Bernoulli model, if the average read rate is p^{avg} for each tag in each epoch in *W*, then the probability that we miss a reading from tag *i* over *W* is $(1 - p^{avg})^W$. Setting the least probability for tag been observed in the window to be δ, then the probability to ensure that tag *i*

been observed is $1-\delta$, that is $(1-p^{avg})^W \le \delta$. So the least size of window is $w^* = \ln(\frac{1}{\delta}) / p^{avg}$, which guarantees completeness.

Because the mobile tags are detected in the preprocessing, we just need to focus on completeness. In addition, w^* is set to infinite when $p^{avg} = 0$. This is not permitted, so we keep the window size unchanged. It is worthy to mention that it adopts a mid-window slide point. In other words, it produces readings with an epoch value corresponding to the midpoint of the window. This greatly reduces the output data.

4.3. SWKF

The Kalman Filter and sliding window are introduced above. In the Kalman Filter model, when the missed read rate is high, the prediction accuracy will be greatly reduced. Then just relying on the Kalman Filter does not correctly reflect the real data (see from figure 2), the cleaning effect is bad. Besides, it produces large amounts of output data.

For the sliding window, the tags that are detected far away from the reader with a low-probability will force it to use larger window to guarantee completeness. But they can cause problems in environment where tags are mobile. For in the multi-tag case, a similar reading results in an overly large contribution to the overall count estimate, and thus a large over-estimation error. Also, when the missed read rate is high, the cleaning effect is bad. Algorithm 1 is the pseudo-code.

Algorithm 1 SWKF
Require: Completeness confidence δ
1: $w \leftarrow 1$
2: While ($getNextEpoch()$) do
3: $kal\_process()$
4: $processWindow(W)$
5: $w^* \leftarrow completeSize(p^{avg}, \delta)$
6: if ($w^* > w$) then
7: $w \leftarrow \max\{\min\{w+2, w^*\}, 1\}$
8: end if
9: end while

SWKF runs a sliding-window aggregate for each observed tag i. The window size is initially set to one epoch, and then adjusted dynamically based on observed readings. During each epoch, for tag i, it utilizes Kalman Filter for preprocessing ($kal\_process()$), which removes interference to some extent, detects and filters the mobile tags. After preprocessing, SWKF processes the reading of tag i inside the window W. The processing includes estimating the required model parameters for tag i (e.g., p^{avg}) as well as emitting an output reading for tag i if there exits at least reading within the window. Then, SWKF consults it binomial-sampling model to determine the number of epochs needed to guarantee completeness ($completeSize(p^{avg}, \delta)$). If the required w^* exceeds the current window size $w = |W|$, SWKF increases the current window size ($\max\{\min\{w+2, w^*\}, 1\}$).

5. Results and Analysis

5.1. Experimental Setup

In our experiments, the hardware environment: 2.61G Athlon dual-core CPU, 1G Memory, 320G Hard Disk. The software environment: Windows XP operating system, Oracle

11g Enterprise Edition, Matlab and PL/SQL language. Experiments are designed to measure the algorithms SMURF, KAL_RFID proposed in [6] and SWKF proposed in this paper.

In order to be more convincing, we collect data in the physical environment (Tiger RF1001 UHF Reader) for 150 read cycles.

5.2. Evaluation Model

In this paper, we assess the space cost and accuracy, and give the standard definition of accuracy.

Definition 3 (Accuracy) Given two data sets, the real data set D_r and cleaned data set D_c. In a time period T, the accuracy of the data can be expressed as:

$$P_A(T) = D_r(T) \cap D_c(T) / D_r(T) \quad (8)$$

5.3. Evaluation Model

This section analyses the reasonableness and the cleaning effect of SWKF. Firstly, the reasonableness is presented. Least squares estimation is used in Kalman Filter prediction.

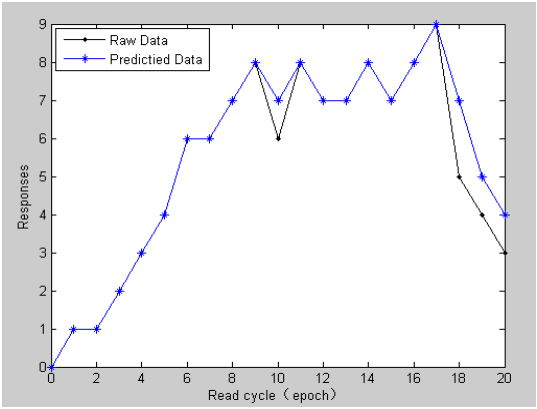

Figure1 Ideal, Kalman filter

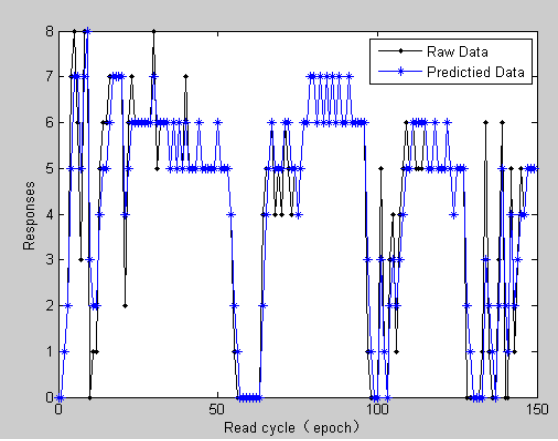

Figure2 Preprocess instance in the real-world environment

Figure 1 shows, in the ideal case, the least squares method makes the data close to the real. In order to be closer to reality, we preprocess the collected data in real-world environment mentioned above. Figure 2 shows that the least squares method can correct read rate (response/10) to a certain extent, but the read leakage rate is still large, that is to say, the data needs to be further processed. The Kalman Filter preprocessing can effectively detect mobile tags, and sliding window is difficult to do this. However, the ability to detect mobile tags is a very important factor that affects the accuracy of sliding window. So the combination of Kalman Filter and sliding window is a workable option, which indicates that SWKF is reasonable.

In order to verify the accuracy of SWKF, it is compared with KAL_RFID and SMURF. Figure 3 shows that the accuracy of SWKF is higher than that of SMURF and KAL_RFID. Although the read leakage rate increases, the accuracy of SWKF is improved.

To show the accuracy of the algorithm more effectively, we clean the data collected in real application environment with SWKF. The results are shown in Figure 4. In the 10th cycle, SMURF has false negative. In the 60th and 100th cycle, tags move dynamically, the read rate is reduced to 0, SMURF increases the window size, and causes false positive. During 130 to 140 cycles, the window size of SMURF changes frequently, and causes large errors. In the first 130 cycles, KAL_RFID handles better, but it has false positive and negative during 130-140 cycles when the tags move frequently (read rate is about 0.3). Because of the preprocessing of Kalman Filter, the read rate is close to the true in the first 130 cycles for SWKF. After the

smoothing of sliding window, it works well. But it also has some false during 130-140 cycles. However, compared to KAL_RFID, it has some improvement.

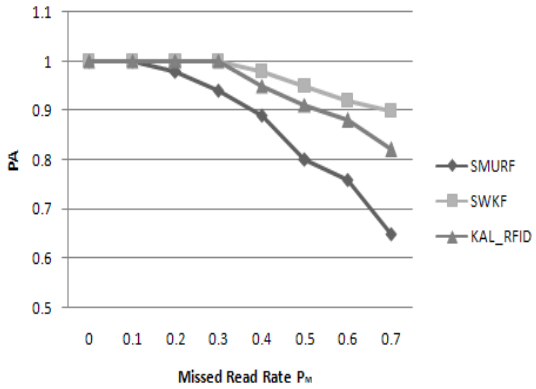

Figure 3. Accuracy

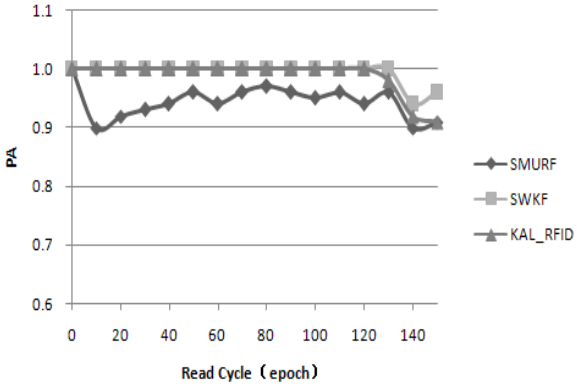

Figure 4. Cleaning instance in the real-world environment

5.4. Space Cost Analysis

This section analyzes the space cost of the algorithm. Figure 5 shows that the number of data output and input is almost the same for KAL_RFID in 150 read cycles. Because the preprocessing filters the mobile tags, and then effectively prevents the sudden increase of the sliding window, so the space cost for SWKF is larger than SMURF. When the number of data increases or the tag moves frequently, SWKF will need much more space than SMURF.

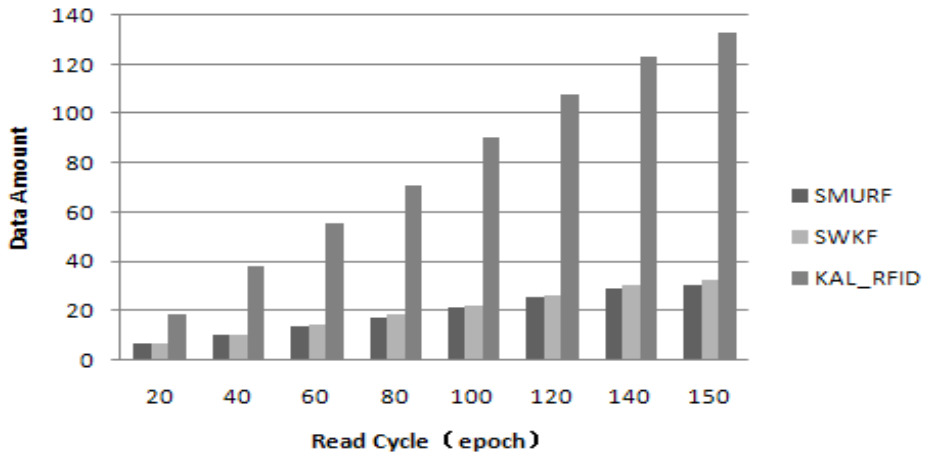

Figure 5. Space Performance

6. Conclusion

RFID technology has a wide application, and also raised new challenges to data cleaning, which has become an important field of RFID research. According to the characteristics of RFID data, we propose a data cleaning algorithm, which utilizes the Kalman Filter to preprocess the RFID data, and then with sliding window for further processing. Experiments show that it has a good cleaning effect in complex environment.

References

[1] Wang Shao-hui, Liu Sujuan. Efficient Passive Full-disclosure Attack on RFID Light-weight Authentication Protocols LMAP++ AND SUAP. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(6): 1458-1464.

[2] Iswanjono, Bagio B, Kalamullah R. An glgorithm for predicting the speed of traffic light violators. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2011; 9(1): 55-64.

[3] Barnaby J. Feder. Despite Wal-Mart's Edict, Radio Tags Will Take Time. New York Times. 2004.

[4] Jeffrey SR, Alonso G, Franklin MJ, Hong W, Widom J. *A pipelined framework for on line cleaning of sensor data streams*. In: Liu L, Reuter A, et al., eds. Proc. of the 22nd Int'l Conf. on Data Engineering. Atlanta: IEEE Computer Society, 2006. 140−142

[5] Jeffery SR, Garofalakis M, Franklin MJ. *Adaptive cleaning for RFID data streams*. In: Dayal U, Whang KY, et al., eds. Proc. of the 32nd Int'l Conf. on Very Large Data Bases. Seoul: ACM, 2006. 163−174.

[6] Bai Y, Wang FS, Liu PY. Efficiently Filtering RFID data streams. In: Proc. of the 1st Int'l VLDB Workshop on Clean Databases. Seoul: Morgan Kaufmann Publishers, 2006. 50−57.

[7] Gonzalez H, Han J, Shen X. *Cost-Conscious cleaning of massive RFID data sets*. In: Proc. of the 23rd Int'l Conf. on Data Engineering. Istanbul: IEEE Computer Society, 2007. 1268−1272.

[8] Khoussainova N, Balazinska M, Suciu D. *Towards correcting input data errors probabilistically using integrity constraints*. In: Chrysanthis RK. Proc. of the 5th ACM Int'l Workshop on Data Engineering for Wireless and Mobile Access. Chicago: ACM, 2006. 43−50.

[9] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, and J. Widom. Declarative Support for Sensor Data Cleaning. In Pervasive, 2006.

[10] EPCGlobal Reader Protocol Standard 1.1. http://www.EPCglobalinc.org. 2006.

[11] Lingyong Meng, Fengqi Yu. *RFID Data Cleaning Based on Adaptive Window*. Proc of the 2nd International Conference on Future Computer and Communication. Wuhan, China, 2010, 746-749.

[12] Wang Yan，Song Bao-yan. Cleaning Method of RFID Data Stream Based on Kalman Filter. *Journal of Chinese Computer Systems*. 2011; 32(9): 1794-1799.

[13] Wang Yan，Shi Xin, Song Bao-yan. RFID Data Cleaning Method Based on Pseudo Event. Journal of Computer Research and Development. 2009; 46(Suppl): 270-274.