Outlier Detection in Cold-chain Logistics Temperature Monitoring

Weidong Zhao[1], Weihui Dai[2], Shangchen Zhou[1]
[1]*School of Software, Fudan University,*
[2]*School of Management, Fudan University,*
220, Handan Road, Shanghai, 200433, P. R. China, phone: +86 21 25011241
whdai@fudan.edu.cn

***Abstract*—In the field of cold chain logistics, the key point is the real time control of temperature. Thus the failure of synchronous temperature monitoring is the bottleneck of cold-chain temperature monitoring. Targeting at the real-time features of synchronous temperature monitoring, this paper discusses some issues about RFID technology applied to outlier detection. Through comparing differing feasible RFID data mining methods, along with the requirements of the cold chain temperature monitoring, we put forward the improved QOD (quick outlier detection) algorithm by clustering based on data stream. After that, we prove that QOD algorithm's performance can be improved after optimization and compared it with several other related methods in accuracy, memory consumption.**

***Index Terms*—Remote monitoring, radiofrequency identification, temperature sensors.**

I. INTRODUCTION

Cold chain was introduced by Albert Barrier and O. A. Ruddich in 1894. But it had not caught the public's attention until the 1940s. Some scholars, including Dne Ouden and Zuurbier, put forward the conception of food supply chain in 1996. Currently, most corporations use related technologies to ensure the safety of frozen food in this field [1]. However, the technologies are mainly put in practice in the production of some aquatic food, meat and canned food while very rarely applied in the transportation and circulation stages, which leads to its inability to ensure the quality of frozen food in transportation. Marija et al. explored the cold chain management of temperature sensitive products [2]. By monitoring the temperature of frozen products, logistics in the transportation stage can use the appropriate packaging and equipment to ensure product quality. Among all the factors of cold chain logistics, the temperature is the most direct and easiest one to deal with, so the safety and quality of cold chain logistics should focus on the temperature monitoring.

In addition to transferring real-time data, RFID also enjoys some advantages. Using RFID temperature monitoring technology, we are able to carry on the real-time control of the temperature as well as trace the change of temperature. Due to the overwhelming data provided by temperature monitoring of RFID cold chain logistics, we need some effective data stream mining methods to process it.

RFID cold chain real-time temperature monitoring aims at finding those temperature points, which are significantly different from others. Outlier mining method concerns itself about only a small portion of data, which is always treated as noise and ignored. Compared with it, other data stream mining methods have remarkable drawbacks in some aspects. In this way, outlier mining may gracefully meet the requirements of RFID cold chain real-time temperature monitoring [3].

Outlier analysis includes methods based on statistics, density, clustering and deviation. Among them, statistics can be further divided into methods based on distribution and depth of data [4], [5]. There are many outlier mining methods based on the distribution. Yamanishi et al. used a Gaussian mix model to calculate a score for data on the basis of the change of the model as a statistical representation of normal behaviours [6]. The method based on the depth can avoids the problem of distribution matching, and allows to handle multidimensional data objects. But in fact, the method for a large data set, for example more than four-dimensional is unrealistic. The existing methods based on depth provide acceptable performance only when the dimension is smaller than 2 [7].

Since RFID cold chain outlier mining algorithm deals with the temperature data which is one-dimension and the real-time temperature data satisfy the normal distribution, we choose the method based on statistics.

II. QUICK OUTLIER DETECTION ALGORITHM

As to the design and implementation of outlier mining, we have put forward the algorithm QOD(quick outlier detection) based on the definition of outliers [8].When using outlier mining method based on the distribution to process the data stream, we have to implement it through an iterative manner. So does QOD. Thus it seems to be slow. Another defect may be the inability to run efficiently in limited memory, so when the amount of data is too huge, we probably have not enough memory. Aiming at solving these

Manuscript received March 19, 2012; accepted May 29, 2012.
This work was supported partly by the Natural Science Foundation of China (71071038).

problems, this paper optimizes QOD from two aspects.

III. Optimization of QOD Algorithm

The optimization methods of QOD proposed include pruning [8] and micro-clustering. We focus on the improvement based on the micro-clustering in this paper.

First, we use the fractal normal distribution theory to prune the data used by QOD so as to remarkably improve its performance [8].

Let the object set $X = \{x_1, x_2..., x_n\}$. According to the definition of outliers based on the normal distribution, for an attribute r of certain object $x_i \in X$, if

$$| \frac{x_{ir} - \mu_r}{\sigma_r} | \geq 3 , \quad (1)$$

then x_i is the outlier, that is, the data object whose attributes deviate from the average value by over *3* is an outlier. In contrast, other objects that do not meet this condition can not be outliers. The local outlier measure of x_i is defined as $LO(xi) = \frac{1}{|N(xi)|} \sum_{i=1}^{|N(xi)|} D(xi,xj), xj \in N(xi)$, where $N(x_i)$ is the local neighbors of x_i, $|N(x_i)|$ represents the number of x_i's local neighbors. Similarly, if the maximal local outlier measure $MaxLO(N+(x_i))$ in a local neighborhood of a certain object y_i does not deviate from the approximate average of the outlier measure, we suppose that such neighborhood can not contain outliers [8]. Herein, $N+(x_i) = N(x_i) \cup \{x_i\}$.

In light of the basic idea of pruning discussed above, we can obtain the following properties.

For a certain object $x_i \in X$, if $MaxLO(N_+(x_i)) < C \cdot GAD$, then the local neighborhood $N+(x_i)$ can not contain any outlier, thus the entire local neighborhood of x_i can be pruned from the data set. Here, global approximate outlier measure *GAD* is the approximate average of the attribute r in the whole data set [8].

For a certain object $x_i \in X$, if $LO(x_i) < C \cdot GAD$ in its local neighborhood, then object x_i can not be the outlier, so we can prune it from the data set. Here $LO(x_i)$ is Local outlier measure of x_i.

Based on these properties, we can prune some objects that can not be outliers from the data set, thereby reducing running time along with memory used to store those data objects.

The essence of stream data is that data continuously arrives over time, so it is impossible to get all the data at one blow. Stream data mining needs to free memory after scanning input data; otherwise with time elapsing, it will probably run out of memory. The method discussed above does not take into consideration the properties of stream data, therefore it can only check for some exceptional cases in the local neighborhood and lose the possibility to discover some global exceptions.

A. Micro-clustering

In order to cluster stream data effectively, some methods have been employed: compute and store the summary information of historical data so as to alleviate the limitation of memory and quickly respond; use divide and conquer policy, divide the data into blocks according to their arriving time, then compute the summary of those blocks, finally merge all this summary information; incremental update of input data stream; imposing micro and macro clustering analysis.

Since we aim at compressing data and reducing the computation efforts while producing as little errors as possible in this paper, we only obtain micro-clusters through the micro-clustering technology and only store the statistical information of every micro-cluster.

In QOD, there is no need to compute the weighted distance with each original data object but with each micro-cluster when computing the local outlier measure of a certain object. Although probably losing some information in data through micro-clustering, we can ensure the accuracy by enhancing the number of micro-clusters.

B. Micro-clustering optimization

Several necessary definitions are given as follows.

Definition 1. $C = \{x_1, x_2... x_n\}$, is the object set that satisfy the requirement that

$$\sum_r \sqrt{\frac{\sum_{i=1}^{n} (x_{ir} - \overline{x_r})^2}{n}} < c , \quad (2)$$

where c is the cohesion factor, r is a property of those objects, $\overline{x_r}$ is the average value of r of a certain micro cluster.

Since

$$\sum_{i=1}^{n} (x_{ir} - \overline{x_r})^2 = \sum_{i=1}^{n} x_{ir}^2 - n\overline{x}^2 , \quad (3)$$

we only need store the related data objects of each micro-cluster, and thereby save memory and reduce computation time.

Definition 2. Micro-clustering outlier measure of a certain object is represented as

$$GO(x_j, \overline{C}_i) = \frac{1}{m} \sum_{i=1}^{m} D(x_j, \overline{C}_i) , \quad (4)$$

where m is the current number of micro-clusters, $\overline{C_i} = \frac{\sum_{x \in C} x_r}{n_i}$, n_i is the number of elements in C_i.

Definition 3. Suppose the micro-cluster set $A = \{C_1, C_2,...,C_m\}$, the average is $\bar{C}_r = \frac{\sum_{i=1}^{m} C_{ir}}{m}$. If $GO(C_i, \overline{C}) > c$, then the micro-cluster C_i is outliers.

The improved QOD algorithm associated with micro-clustering is described as follows:

Step 1. Initialize the micro-cluster set to be empty.

Step 2. Scan the input object xi, suppose we have got the micro-cluster set A = {C1,C2, …,Cm} through micro-clustering, if A is empty, then xi forms a micro cluster itself, and we add it to A, return to Step 1.

Step 3. For each micro cluster in the current A, we suppose that we add xi to one of the micro-clusters and compute the variance. If one of the minimal variance

satisfies the merit of cohesion, then we add xi to this cluster and update the set, return to step 1.

Step 4. If it does not satisfy the merit of cohesion but meet that of micro-clustering outliers, we define it as a single micro-cluster and add it to A, return to step 1.

Step 5. If it does not satisfy the merit of micro-clustering outliers, then it is outlier, return to step 1.

Although some outliers may not be detected through micro-clustering, we can make up for the method through the detection of outlying clusters, which are the micro-clusters probably including outliers. The system can notify us when finding outlying cluster and then underlying reasons can be found.

Also the micro-clustering method can be combined with the pruning algorithm, which further improves the performance.

The algorithmic complexity of QOD based on micro-clustering depends on the number of micro-clusters. The pruning method has the complexity of $O(kn\log_s n)$. So when associated with pruning algorithm, the complexity is $O(km\log_s m)$, m is the number of micro-clusters. As the number of clusters is much smaller than the number of data objects in general conditions, the time used to run QOD with micro-clustering optimization is much less than QOD with pruning.

IV. EXPERIMENTS

The dataset used in our experiments is from the collection of temperature data in cold chain logistics project conducted by the research group of Columbia University and a large logistics enterprise in U.S.A. jointly. It contains 8124 groups of sample data. The temperature data is content-rich and well formatted.

Although the data mentioned above is not in the format of RFID data, there are many similarities between these two kinds of data after experimental data has been initialized and pre-processed. Therefore, the processed temperature data can be used. During the experiments, we randomly select 500 groups of data before each iterative analysis step so as to avoid systematic errors caused by the small amount of sample data.

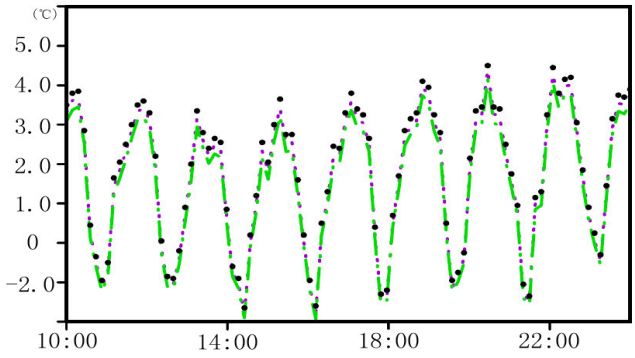

Fig. 1. Fragment of sample temperature data over time.

Fig. 1 shows the curves of temperature data in a slide window from different cold chain models. It records a series of cold chain real-time temperature data of which the temperature is required to be bounded from -21℃ to -15℃. Obviously, we can see there are some outliers in the temperature curves that are out of the common scope.

Considering the rapid change and huge amount of data, we compare the error rate, memory consumption during the running period and average running time based on the sample data among several algorithms including QOD, QOD with pruning, QOD with clustering, QOD with pruning and clustering and slide window [9]. The algorithm which has better overall performance in the three check points is considered as a more effective one with stronger comprehensive property.

The data set used is generated and processed by MATLAB, including about 40000 data objects.

Table I shows the error rates of the algorithms mentioned above. The error rate of the optimized QOD algorithms is almost the same as that of the pure QOD. The reason is that there are no great changes in local outlier measure factor after pruning. Therefore, the optimization can not significantly impact the accuracy. On the other hand, the error rate of QOD is much lower than any other algorithm, because QOD does not use the regional approximation while others do. The sliding window algorithm can not quantify and describe diversification, and it is not sensitive to outliers in smooth diversification. Obviously, the pruning and clustering method both discard some data, so their performance is a little worse than the original QOD.

TABLE I. ERROR RATES OF DIFFERENT ALGORITHMS.

Algorithm	**Error rate(%)**
QOD	5.77(±0.63)
QOD with pruning	5.79(±0.64)
QOD with clustering	5.77(±0.66)
QOD with pruning and cluster	5.83(± 0.67)
Slide Window	18.30(±0.65)

Fig. 2 shows the memory consumption growth of the algorithms when the number of data objects is increased. The result shows the slide window method use less memory than others, for the reason that it only takes a piece of data into consideration. So the size of data has little effect on the memory that the slide window method uses. On the other hand, memory used by QOD with clustering is much less than that used by QOD or QOD with pruning. Because memory used by QOD with clustering is only associated with the number of clusters, which is only relative to the distribution of data, so it does not consume much space generally. QOD and QOD with pruning consume a lot of memory, because they must keep all data in the memory and pruning has little effect on memory usage, thereby can not improve the performance. If the size of dataset is too large, then we need swap some data into storage disk to lower cost.

Moreover, QOD and QOD with pruning are significantly impacted as the size of dataset is increased rapidly while the slide window method and QOD with clustering do not have large fluctuation. The reason is that the slide window dynamically moves over time and releases unused resources, and QOD with clustering enjoys a small amount of clusters in general conditions thereby saving memory in contrast to pure QOD and QOD with pruning.

Fig. 3 shows running time of several different algorithms

as the number of data objects is increased.

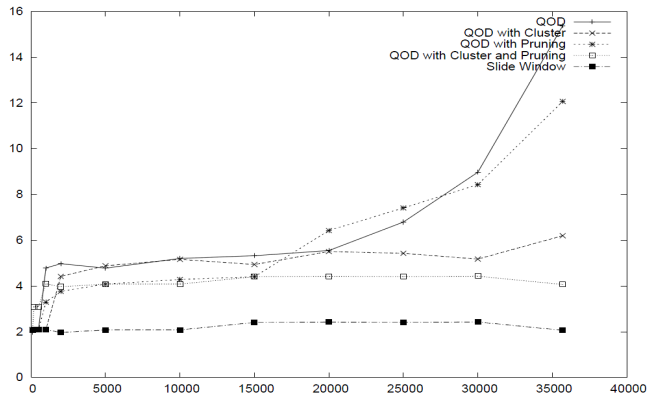

Fig. 2. Memory consumption of different algorithms.

We can see that the slide window method is fastest, since it only compares the data in the window. QOD with pruning is a little faster than QOD because for a part of points it omits a lot of calculations. Therefore, the algorithm does not need iterative computation for every RFID data object. It results from the fact that extra space storing reusable information is generated after pruning.

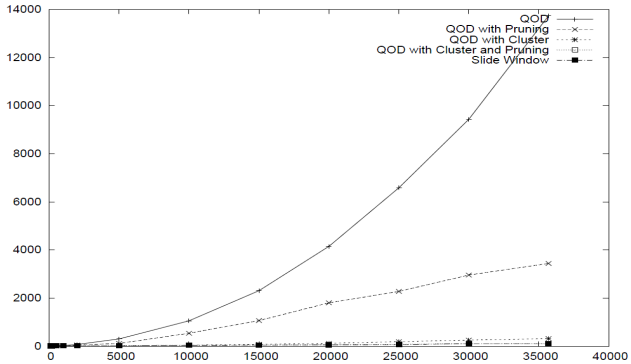

Fig. 3. Running time of different algorithms.

V. CONCLUSIONS

This paper focuses on the improvement of QOD. The pruning method runs faster. In contrast, the clustering method decreases memory used, and also becomes efficient. In the future, we will resolve outliers which are detected, construct rule database with domain knowledge, and use the database to analyze outliers. Moreover, the accuracy of the method needs further improvement.

ACKNOWLEDGMENT

Many thanks to Hongzhi Hu for her assistant work to the corresponding author Weihui Dai of this article.

REFERENCES

[1] V. M. D. Neil, *Tracking weakness links in cold chain.* Berkeley CA: University of California Press, 2006, p. 342.

[2] B. Marija, et al., "Stability of perishable goods in cold logistic chains", *International Journal of Production Economics,* vol. 93–94, no. 8, pp. 345–356, 2005.

[3] D. Hawkins, *Identification of outliers.* London: Chapman & Hall, 1980, p. 188. [Online]. Available: http://dx.doi.org/10.1007/978-94-015-3994-4

[4] E. M. Knorr, V. Tucakov, "Distance-based outliers: algorithms and applications", *The VLDB Journal,* vol. 8, no. 3–4, pp. 237–253, 2000. [Online]. Available: http://dx.doi.org/10.1007/s007780050006

[5] L. Breiman, J. H. Friedman, R. A. Olshen, et al., *Classification and regression trees.* New York: Chapman &Hall, 1984, p. 368.

[6] T. Yamanishik, "Discovering outlier filtering rules from unlabeled data-combining a supervised learner with an unsupervised learner", in *Proc of the KDD01,* New York, 2001, pp. 389–394.

[7] T. Yamanishik, et al., "On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms", in *Proc. of the KDD00*, ACM Press, 2000, pp. 320–324.

[8] Z. Weidong, S. C. Zhou, Y. M. Sun, "Research on temperature control based on outlier mining in RFID cold chain", *Computer System Applications,* vol. 19, no. 11, pp. 1661–1667, 2010.

[9] W. Bin, Y. Xiaochun, W. Guoren, et al., "Outlier detection over sliding windows for probabilistic data streams", *Journal of Computer Science and Technology,* vol. 25, no. 3, pp. 389–400, 2010. [Online]. Available: http://dx.doi.org/10.1007/s11390-010-9332-2