Estimating Missing Data in Data Streams*

Nan Jiang and Le Gruenwald

The University of Oklahoma
School of Computer Science
Norman, OK, 73019, USA
{nan_jiang, ggruenwald} @ou.edu

Abstract. Networks of thousands of sensors present a feasible and economic solution to some of our most challenging problems, such as real-time traffic modeling, military sensing and tracking. Many research projects have been conducted by different organizations regarding wireless sensor networks; however, few of them discuss how to estimate missing sensor data. In this research we present a novel data estimation technique based on association rules derived from closed frequent itemsets generated by sensors. Experimental results compared with the existing techniques using real-life sensor data show that closed itemset mining effectively imputes missing values as well as achieves time and space efficiency.

1 Introduction

Many research projects have been conducted by different organizations regarding wireless sensor networks; however, few of them discuss how to estimate the sensor data that are missing because they are lost or corrupted or arrive late when being sent from sensors to servers. Traditional methods to handle the situation when data is missing are to ignore them, make sensors to send them again or use some statistical methods to perform the estimation. As we discuss in Section 2, these methods are not specially suited for wireless sensor networks.

In this paper, we propose a data estimation technique using association rule mining on stream data based on closed frequent itemsets (CARM) to discover relationships between sensors and use them to compensate for missing data. Different from other existing techniques [4-6, 10, 12], CARM can discover the relationships between two or more sensors when they have the same or different values. The derived association rules provide complete and non-redundant information; therefore they can improve the estimation accuracy and achieve both time and space efficiency. Furthermore, CARM is an online and incremental algorithm, which is especially beneficial when users have different specified support thresholds in their online queries.

The remainder of this paper is organized as follows. Section 2 describes the data missing problem and reviews the existing data estimation solutions. Section 3 discusses the definitions of terms used in the paper. Section 4 presents the proposed online data estimation algorithm based on the discovered closed frequent itemsets. Section 5 depicts the performance evaluation comparing the proposed algorithm with the existing techniques using real-life traffic data. Finally, Section 6 concludes the paper.

* This research is partially supported by the NASA grant No. NNG05GA30G and a research grant from the United States Department of Defense.

2 Related Works

Many articles have been published to deal with the missing data problem, and a lot of software has been developed based on these methods. Some of the methods totally delete the missing data before analyzing them, like listwise and pairwise deletion [16], while some other methods focus on estimating the missing data based on the available information. The most popular statistical estimation methods include mean substitution, imputation by regression [3], hot deck imputation [7], cold deck imputation, expectation maximization (EM) [10], maximum likelihood [2, 9], multiple imputations [11, 13], and Bayesian analysis [5]. However, a number of problems arise when applying them to sensor networks applications. First, none of the existing statistical methods answers the question that is critical to data stream environments: how many rounds of information should we use in order to get the associated information for the missing data estimation? Second, it is difficult to draw a pool of similar complete cases for a certain round of a certain sensor when it needs to perform the data estimation, which makes some statistical methods difficult to use. Third, since the missing sensor data may or may not be related to all of the available information, using all of the available information to generate the result as described in some of the statistical methods would consume unnecessary time. And fourth, sensor data may or may not Miss At Random (MAR), which makes it unfavorable to use those statistical methods that require the MAR property.

In [6], the authors proposed the WARM (Window Association Rule Mining) algorithm for estimating missing sensor data. WARM uses association rule mining to identify sensors that report the same data for a number of times in a sliding window, called related sensors, and then estimates the missing data from a sensor by using the data reported by its related sensors. WARM has been reported to perform better than the average approach where the average value reported by all sensors in the window is used for estimation. However, there exist some limitations in WARM. First, it is based on 2-frequent itemsets association rule mining, which means it can discover the relationships only between two sensors and ignore the cases where missing values are related with multiple sensors. Second, it finds those relationships only when both sensors report the same value and ignores the cases where missing values can be estimated by the relationships between sensors that report different values.

In view of the above challenges, in this paper we propose a data estimation technique, called CARM (Closed Itemsets based Association Rule Mining), which can derive the most recent association rules between sensors based on the current closed itemsets in the current sliding window. The definition of closed itemsets is given in Section 3 where we describe the notations that are used throughout this paper.

3 Definitions

Let $D = \{d_1, d_2, \ldots, d_n\}$ be a set of n item ids, and $V = \{v_1, v_2, \ldots, v_m\}$ be a set of m item values. An item I is a combination of D and V, denoted as $I = D.V$. For example, $d_n.v_m$ means that an item with id d_n has the value v_m. A subset $X \subseteq I$ is called an itemset. A k-subset is called a k-itemset. Each transaction t is a set of items in I. Given a set of transactions T, the support of an itemset X is the percentage of transactions that contain X. A frequent itemset is an itemset the support of which is above or equal to a user-defined support threshold [1].

Let T and X be subsets of all the transactions and items appearing in a data stream D, respectively. The concept of closed itemset is based on the two following functions, f and g: $f(T) = \{i \in I \mid \forall t \in T, i \in t\}$ and $g(X) = \{t \in D \mid \forall i \in X, i \in t\}$. Function f returns the set of itemsets included in all the transactions belonging to T, while function g returns the set of transactions containing a given itemset X. An itemset X is said to be closed if and only if $C(X) = f(g(X)) = f \bullet g(X) = X$ where the composite function $C = f \bullet g$ is called Galois operator or closure operator [14].

From the above discussion, we can see that a closed itemset X is an itemset the closure C(X) of which is equal to itself (C(X) = X). The closure checking is to check the closure of an itemset X to see whether or not it is equal to itself, i.e., whether or not it is a closed itemset.

An association rule $X \rightarrow Y$ (s, c) is said to hold if both s and c are above or equal to a user-specified minimum support and confidence, respectively, where X and Y are sensor readings from different sensors, s is the percentage of records that contain both X and Y in the data stream, called support of the rule, and c is the percentage of records containing X that also contain Y, called the confidence of the rule. The task of mining association rules then is to find all the association rules among the sensors which satisfy both the user-specified minimum support and minimum confidence.

4 Data Estimation Algorithm based on Closed Frequent Itemsets

In this section, we present an online data estimation technique called CARM based on a closed frequent itemsets mining algorithm in data streams that we have proposed recently, called the CFI-Stream [8]. We first briefly describe the CFI-Stream data structure called DIrect Update (DIU) tree that is used to compute online the closed frequent itemsets in data streams. Then we discuss how to estimate the missing data based on the association rules derived from the discovered closed frequent itemsets.

A lexicographical ordered direct update tree is used to maintain the current closed itemsets. Each node in the DIU tree represents a closed itemset. There are k levels in the DIU tree, where each level i stores the closed i-itemsets. The parameter k is the maximum length of the current closed itemsets. Each node in the DIU tree stores a closed itemset, its current support information, and the links to its immediate parent and children nodes. Fig.1. illustrates the DIU tree after the first four transactions arrive. The support of each node is labeled in the upper right corner of the node itself. The figure shows that currently there are 4 closed itemsets, C, AB, CD, and ABC, in the DIU tree, and their associated supports are 3, 3, 1, and 2. We assume in this paper that all current closed itemsets are already derived, and based on these closed itemsets, we generate association rules for data estimation. Please refer to [8] for the detail discussion of the update of the DIU tree and the closure checking procedures.

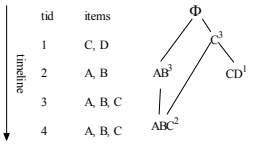

Fig. 1. The lexicographical ordered direct update tree

CARM proceeds in the following manner. First, it checks if there are missing values in the current round of sensor readings. If yes, it uses the current round of readings X that

contains the missing items to find out its closure online. If the rules from X to its immediate upper level supersets satisfy the user specified support and confidence criteria, these upper level supersets are treated as starting points to explore more potential itemsets until CARM estimates all missing sensor data. Following this method, CARM continues to explore and find all closed itemsets that can generate association rules satisfying the users' specified support and confidence criteria. All these closed itemsets are the supersets of the exploration set and have the support and confidence along the path above or equal to the users' specified thresholds.

```
1 X_estimate=ϕ;
2 For all (M ⊆ X)
3    conf_M=1;C_estimate(M, conf_M, X_estimate)
4    If (X_estimate contains all the missing values)
5       break;
6 End for
7 Procedure C_estimate(X, Conf_x, X_estimate){
8    X_new=ϕ;
9    If (X=Closure(X))
10      For all (Y⊃X and Y∈C and Y = min(X))
11              Conf_y=Conf_x*Support(Y)/Support(X);
12              X_new = X_new∪(Y/X_estimate)
13      End for
14      For all (I∈X_new)
15         For all (Z⊃X and Z=min(Z))
16                    If(I∈Z)
17                    Conf_I=Conf_Z;
18         End for
19         If (Support(I∪X)>S_specify and Conf_I>C_specify)
20                          S_(I).V_I =S_(I).V_I +Conf_I*V_I
21       End for
22       If(X_new doesn't contain all missing sensor data)
23         For all (X'⊃X and X'∈C and X'= min(X))
24               Call C_estimate(X', Conf_x', X_estimate∪ X_new)
25       End if
26    Else
27      X_c=Closure(X); X_new=X_c/X; Conf_Xc=1;
28      If(Support(X_c)> S_specify )
29         For all (J∈ X_new)
30              Conf_J = Conf_Xc; S_(J).V_J =S_(J).V_J +Conf_J*V_J;
31      End if
32      If(X_new doesn't contain all missing sensor data)
33              Call C_estimate(X_c, Conf_Xc, X_new)
34   End if
35 End procedure
```

Fig. 2. The CARM online data estimation algorithm

CARM generates the estimated value based on the rules and selected closed itemsets, which contain item value(s) that are not included in the original readings X. It weights each rule by its confidence and calculates the summation of these weights multiplied with their associated item values as the final estimated result. These item values can be

expected as the missing item values with the support and confidence values equal to or greater than the users' specified thresholds. In this way, CARM takes into consideration all the possible relationships between the sensor readings and weights each possible missing value by the strength (confidence) of each relationship (rule). This enables CARM to produce a final estimated result near the actual sensor value based on all of the previous sensor relationships information. We show the CARM algorithm in Fig. 2, where X is the itemset in the current round of sensor readings, Y represents all supersets of X, $Conf_y$ represents the strength of the rule from itemset X to Y, Support(X) represents X's support, Closure(X) is the closure of itemset X in the current transactions, Min(X) represents X's immediate upper level supersets in the DIU tree, C represents all closed frequent itemsets, $S_{(I)}$,V_I represents the value V_I of sensor id $S_{(I)}$, $X_{estimate}$ represents the returned estimation itemset which contains the sensor ids with missing values in the current round of readings of stream data and their corresponding estimated values, $S_{specify}$ represents the user specified support, and $C_{specify}$ represents the user specified confidence.

5 Experimental Evaluations

Several different simulation experiments are conducted comparing CARM with four existing statistical techniques: Average Window Size (AWS), the Simple Linear Regression (SLR), the Curve Regression (CE), and Multiple Regression (MR), and with WARM, a data estimation algorithm in sensor database [6].

As shown in Fig. 3(a), the experiment results show that CARM gives the best estimation accuracy, followed by WARM and AWS. The regression approaches perform worse than WARM, CARM and AWS. The main reason of this might be that they only based on the regressions between the neighbor sensor readings, while CARM and WARM discover all of the relationships between the existing sensors. CARM provides better estimation accuracy than WARM because the association rules in CARM are derived from a compact and complete set of information, while those in WARM are derived from only the 2-frequent itemsets in the current sliding window.

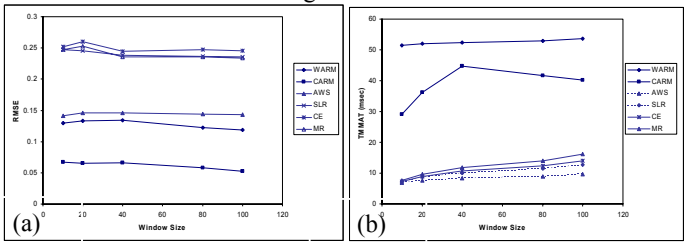

Fig. 3. RMSE and TMMAT for AWS, SLR, CE, MR, WARM and CARM approaches

In terms of TMMAT, which is the time for performing all main memory accesses required for updating the associated data structures and estimating missing values per round of sensor readings, as shown in Fig. 3(b), CARM is outperformed by all other four statistical approaches, but it is still very fast comparing with the cases in which sensors must resend the missing data, and is faster than WARM. The TMMAT of WARM increases slightly when the window size increases since the information in WARM is stored in the cube data structures, and the time needed to process this information increases when the size of the cube increases. For CARM, the TMMAT first increases as

the number of transactions increases since the number of closed itemsets that are newly discovered increases.

In terms of Memory Space, CARM is outperformed by all other four statistical approaches, but it still requires far less memory space than that provided in a contemporary computer. The needed memory space in CARM is much lower than that in CARM because the tree data structure used in CARM stores only the condensed closed itemset information while the cube data structures in WARM store the sensor readings of all sensors and the supports of pairs of sensors in the current sliding window.

6 Conclusions

In this paper we proposed a novel algorithm, called CARM, to perform data estimation in sensor network databases based on closed itemsets mining in sensor streams. The algorithm offers an online method to derive association rules based on the discovered closed itemsets, and imputes the missing values based on derived association rules. It can discover the relationships between multiple sensors not only when they report the same sensor readings but also when they report different sensor readings. Our performance study shows that CARM is able to estimate missing sensor data online with both time and space efficiency, and greatly improves the estimation accuracy.

References

1. R. Agrawal, T. Imielinski, A. Swami; Mining Association Rules between Sets of Items in Massive Databases; Int'l Conf. on Management of Data; May 1993.
2. Allison, P. D. Missing data. Thousand Oaks, CA: Sage; 2002.
3. Cool, A. L. A review of methods for dealing with missing data; Annual Meeting of the Southwest Educational Research Association, Dallas, TX. 2000.
4. Dempster, N. Laird, and D. Rubin; Maximum Likelihood from Incomplete Data via the EM Algorithm; Journal of the Royal Statistical Society; 1977.
5. Gelman, J. Carlin, H. Stern, and D. Rubin; Bayesian Data Analysis; Chapman & Hall; 1995.
6. M. Halatchev and L. Gruenwald; Estimating Missing Values in Related Sensor Data Streams; Int'l Conf. on Management of Data; January 2005.
7. Iannacchione, V. G. Weighted sequential hot deck imputation macros. Proceedings of the SAS Users Group International Conference; 1982.
8. N. Jiang and L. Gruenwald, "CFI-Stream: Mining Closed Frequent Itemsets in Data Streams", ACM SIGKDD intl. conf. on knowledge discovery and data mining, 2006.
9. Little, R. J. A., Rubin, D. B. Statistical analysis with missing data; John Wiley and Sons. 1987.
10. G. McLachlan and K. Thriyambakam; The EM Algorithm and Extensions; John Wiley & Sons; 1997.
11. D.Rubin. "Multiple Imputations for Nonresponce in Surveys". John Wiley & Sons; 1987
12. D. Rubin; Multiple Imputations after 18 Years; Journal of the American Statistical Association; 1996.
13. J. Shafer; Model-Based Imputations of Census Short-Form Items; Annual Research Conference, Washington, DC: Bureau of the Census, 1995.
14. R. Taouil, N. Pasquier, Y. Bastide and L. Lakhal; Mining Bases for Association Rules Using Closed Sets; International Conference on Data Engineering; 2000.
15. O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. Altman. "Missing Value Estimation Methods for DNA Microarrays;" Bioinformatics, 17, 2001.
16. Wilkinson & The APA Task Force on Statistical Inference, 1999.