

A Multivariate Model for Data Cleansing in Sensor Networks

Ping Ji

Dept. of Mathematics & Computer Science
John Jay College of Criminal Justice
Computer Science PhD Program, Graduate Center
City University of New York
New York, NY
Email: pji@jjay.cuny.edu

Marcin Szczodrak

Computer Science PhD Program
Graduate Center
City University of New York
New York, NY
Email: mszczodrak@gmail.com

Abstract—In this research, our objective is to detect and repair erroneous (i.e. dirty) data caused by inevitable system problems involving various hardware and software components of sensor networks. Our approach is to leverage on the intrinsic redundancies and correlations among the collected data, as information about a single event of interest in a sensor network is usually reflected in multiple measurement data points. This data correlation can exhibit temporally, spatially, and across different data types. The inconsistency among multiple sensor measurements serves as an indicator for data quality problem. The focus of this paper is therefore to study methods that can effectively identify and correct erroneous data among inconsistent observations based on the correlation structure of various sensor measurement series. We propose a multivariate model to achieve this goal.

Along the line of sensor data cleansing research, a few studies have been conducted during recent years [3], [2], [4], [5]. In our project, we would like to study more efficient mathematical models to accurately identify dirty data and further to propose a data quality scoring model so that the quality of sensor measurements can be “graded” and reported to upper layer fusion algorithms in a timely manner. The evaluation of data quality is especially crucial for sensor networks that are designed to accomplish military missions. Thus, the ultimate outcomes of this research will be “error-corrected” information and a fidelity measure of the information that would be useful for the decision making process serving military applications.

I. A MULTIVARIATE MODEL FOR OUTLIER DETECTION

A. Identify Correlated Data Groups

In this study, our goal is to utilize spatial, temporal or other types of data redundancies among sensors to identify and repair dirty data. To discover the correlation structures, we need to first classify potentially correlated sensors into groups. For instance, in a sensor network presented by Intel Berkeley Lab [1] as shown in Figure 1, for which we will discuss in further detail in Section II, the sensor nodes can be grouped together base on their regional relationship, and each sensor can be classified into multiple regions (i.e., groups). However, for different sensor network applications the correlation structure of sensor nodes can be varied and more complicated than simple geographic classification. Nonetheless, regardless of sensor node grouping method, the *correlation based data cleansing model* that we propose in this paper can be applied to many sensor networks with correlated sensor nodes being grouped together base on application-specific correlation structure.

B. Dirty Data Detection

The next step is to detect whether potential data problem has occurred at a particular time among the data collected from a

chosen group of sensor nodes. To accomplish this, correlated error structure among multiple sensors should be captured. We thus design a multivariate error model, in which we assume the sensor data processes used to obtain covariance structure are in stationary-state. We further define that the measured data at time t of a sensor node i is \hat{x}_{it} , and the true value of the measurement point is x_{it} . Here, the error sequence of the time series model for the measurements of sensor i can be denoted as

$$\epsilon_t^i = \hat{x}_{it} - x_{it} \quad (1)$$

Various error forecast models can be utilized to capture ϵ_t^i . We adopt a simple moving average, MA(q), model in which ϵ_t^i is assumed normally distributed and follows

$$\epsilon_t^i = \theta_t x_{it} + \theta_{t-1} x_{i,t-1} + \dots + \theta_{t-q} x_{i,t-q}. \quad (2)$$

At each time t , therefore, an error vector can be shown as

$$\Delta_t = [\epsilon_t^1, \epsilon_t^2, \dots, \epsilon_t^n] \quad (3)$$

with n being the number of sensors in the chosen group.

To detect problematic time points in a sensor group, we choose to use *steady-state* measurement data points as the training data sequence to obtain multivariate covariance for later *outlier* detection. We define the multivariate error at time t as

$$\omega_t = \Delta_t \cdot R^{-1} \cdot \Delta_t^T \quad (4)$$

in which, Δ_t is defined in Equation (3), and R is the covariance matrix of error sequences as shown below

$$R = \begin{bmatrix} \text{cov}(\epsilon^1, \epsilon^1) & \text{cov}(\epsilon^1, \epsilon^2) & \dots & \text{cov}(\epsilon^1, \epsilon^n) \\ \text{cov}(\epsilon^2, \epsilon^1) & \text{cov}(\epsilon^2, \epsilon^2) & \dots & \text{cov}(\epsilon^2, \epsilon^n) \\ \dots & \dots & \dots & \dots \\ \text{cov}(\epsilon^n, \epsilon^1) & \text{cov}(\epsilon^n, \epsilon^2) & \dots & \text{cov}(\epsilon^n, \epsilon^n) \end{bmatrix}. \quad (5)$$

In Equation (5), ϵ^i represents the error vector of sensor i . Specifically, $\epsilon^i = \langle \epsilon_{1t}^i, \epsilon_{2t}^i, \dots, \epsilon_{Tt}^i \rangle$ with T being the number of data points from sensor i that are used for training the multivariate covariances. After obtaining the multivariate error sequence of ω_t , we compare ω_t with χ^2 distribution with a threshold τ to detect outliers. Specifically, if $\omega_t > \chi^2(n, \tau)$, it is determined that potential dirty measurement exists in data vector $[\hat{x}_{1t}, \hat{x}_{2t}, \dots, \hat{x}_{nt}]$ at time t .

C. Dirty Data Identification

The multivariate error presented in Equation (4) can only be used to detect the existence of dirty data in a group of sensors at a given time. However to identify whether a specific sensor is producing erroneous measurement, we need to further examine each sensor record individually. By assuming dirty data are sparse, that is to assume that in one sensor group at one time there is only one sensor reports dirty data, we may evaluate the *worst* sensor measurement produced at time t by

iteratively deriving a modified multivariate error, $\omega_t^{<i>}$, which excludes the data entry of sensor i that is under investigation. Specifically, we define

$$\omega_t^{<i>} = \Delta_t^{<i>} \cdot R^{<i>^{-1}} \cdot \Delta_t^{<i>^T}, \quad (6)$$

in which $\Delta_t^{<i>}$ and $R^{<i>}$ are Δ_t and R excluding the i -th element respectively. That is,

$$\Delta_t^{<i>} = [\epsilon_t^1, \dots, \epsilon_t^{i-1}, \epsilon_t^{i+1}, \dots, \epsilon_t^n] \quad (7)$$

and

$$R^{<i>} = \begin{bmatrix} \text{cov}(\epsilon^1, \epsilon^1) & \dots & \text{cov}(\epsilon^1, \epsilon^{i-1}) & \text{cov}(\epsilon^1, \epsilon^{i+1}) & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \text{cov}(\epsilon^{i-1}, \epsilon^1) & \dots & \text{cov}(\epsilon^{i-1}, \epsilon^{i-1}) & \text{cov}(\epsilon^{i-1}, \epsilon^{i+1}) & \dots \\ \text{cov}(\epsilon^{i+1}, \epsilon^1) & \dots & \text{cov}(\epsilon^{i+1}, \epsilon^{i-1}) & \text{cov}(\epsilon^{i+1}, \epsilon^{i+1}) & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \quad (8)$$

By iteratively evaluating $\omega_t^{<i>}$ over $i \in [1, 2, \dots, n]$ for any time t where multivariate error is detected, a sensor m that generates the minimum $\omega_t^{<i>}$ can be identified, which indicates x_{mt} is a dirty data point. Here, it should be noticed that a sensor node can be classified into multiple measurement groups, therefore its measurement may be detected dirty multiple times in one period. This information is useful to enhance the confidentiality of dirty data detection, or to justify whether the error detection itself is biased.

II. PERFORMANCE EVALUATION

In our evaluation, we choose to use the data traces collected from a sensor network built at Intel Berkeley Research Lab shown in Figure 1. During this measurement study [1], 54 Mica2Dot sensors were monitored over a 37-days period, with humidity, temperature, light and voltage values being recorded periodically at each sensor. The data was collected using TinyDB in-network query processing system built on the TinyOS platform.

After examining measurement traces of the Intel Berkeley Lab sensors, we observe that data readings in the traces do not always align with sampling intervals (30-31 seconds), and missing data were found both sporadically and in continuous blocks. Therefore, we prepare the data by constructing equal length discrete time series for each node. Specifically, we process the following three steps for each sensor trace:

1. Group sensor data into fixed length bins (e.g., 1 minute);
2. If multiple data points are classified into one bin, take the average of the data values as the value in that bin;
3. If a bin is empty, fill it by time series forecast such as a simple MA-1 model which uses the previous available data point as the value for an empty bin.

We apply our model proposed in Section I on the Intel Berkeley network. To simplify the evaluation, we assume a simple MA(1) model for measurement errors, that is to define $\epsilon_t^i = x_{it} - x_{i,t-1}$. In addition, we use the first 10,000 data points to train covariance matrix R . Our preliminary evaluation results show that the correlation based model can visually identify a majority of dirty data points efficiently. As for the future work, we would like to examine specific quantitative metrics such as false positive and false negative rates derived from the model.

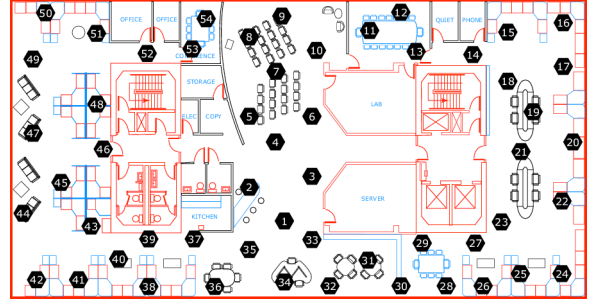


Fig. 1. Intel Berkeley Lab Sensor Network

III. A PHASE-TRANSITION MODEL FOR SENSOR QUALITY

Based on dirty data identification at each sensor node, we can construct a phase-transition model to rate the data quality of a sensor by assigning “quality scores” for it over time. A simple temporal phase-transition model is shown in Figure 2, in which the data quality of a sensor node is rated as “good”, “questionable”, and “bad”.



Fig. 2. A Sample Phase Transition Model for Sensor Quality States

State transition rules, therefore, can be constructed to control the quality rating at each sensor. For instance, some sample state transition rules are: **1.** If a sensor reports non-dirty data for consecutive 5 minutes, then rate the sensor as in “good” data quality state; **2.** If a sensor reports two “dirty” data points over the past 5 minutes, then rate the sensor as in “bad” data quality state; **3.** If none of the above two rules is satisfied, then the sensor is rated as in “questionable” data quality state.

In the future work, we plan to study feasible phase-transition models for evaluating sensor data qualities especially in the scenarios of military applications.

Acknowledgements

Research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

REFERENCES

- [1] Intel Berkeley Lab. <http://db.csail.mit.edu/labdata/labdata.html>.
- [2] E. Elnahrawy and B. Nath. Cleaning and querying noisy sensors. In *Proceeding of International Workshop of Wireless Sensor Networks and Applications (WSNA)*, 2003.
- [3] S. Jeffery, G. Alonso, M. Franklin, W. Hong, and J. Widom. Declarative support for sensor data cleansing. In *Proceeding of International Conference on Pervasive Computing*, 2006.
- [4] R. Rajagopal, X. Nguyen, S. C. Ergen, and P. Varaiya. Distributed online simultaneous fault detection for multiple sensors. In *Proc. of IPSN*, 2008.
- [5] Y. L. Tan, V. Sehgal, and H. H. Shahri. Sensoclean: Handling noisy and incomplete data in sensor networks using modeling. Technical report, University of Maryland. On-line Technical Report, <http://www.cs.umd.edu/~viveks/finalreport.pdf>.