Estimating missing sensor data in Smart Firefighting

Emiliia Dyrenkova[1], Dr. Nalini Venkatasubramanian[2]

***Abstract*— Internet of Things (IOT) sensor networks are important instruments for providing real-time data. Sensor networks have different application: from real-time weather reporting, to water contamination sensing, to fire detection systems. However, such IOT systems are prone to network loss and devices malfunction, especially in the emergency situation. Our task was to define possible algorithms for data estimation that could work reliably in following constraints: limited amount of data available for estimation (real time predicting), increasing amount of missing samples, fire setting data (changing quickly and unpredictably). A common design pattern for the algorithms was created and four algorithms were tested to inspect the performance of nave approaches to data estimation in this settings.**

I. INTRODUCTION

Internet of Things (IOT) sensor networks are important instruments for providing real-time data. Sensor networks have different application: from real-time weather reporting, to water contamination sensing, to fire detection systems. Consider the latter: traditional fire detection devices get activated when a fixed threshold value such as temperature or smoke concentration is reached. Fire detection devices that are connected to the Internet could provide more functionality and data analysis. Smart fire detection networks could automatically notify emergency services, coordinate evacuation and provide real time situation to firefighters. However, such IOT systems are prone to network loss and devices malfunction, especially in the emergency situation. Missing data in the real time analytics could slow down the process. If there is an algorithm to predict missing samples accurately and fast enough, that works with the limited amount of data available, the information obtained from sensor networks would be more complete and important decisions could be better informed.

II. DESIGN

The data was assumed to be represented as a list of samples. The list represents the samples available to the system at the moment. The problem is to be able to estimate any sample from that list using other samples in the list.

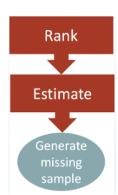


Fig. 1. Process diagram

There were total of four versions of the algorithm developed. All of them consisted of two separate processes: ranking and estimation.

[1]MiraCosta College dyrenkova.emilia@gmail.com

[2]Donald Bren School of Information and Computer Sciences, University of California, Irvine nalini@ics.uci.edu

Ranking means assessing similarity of available samples to the missing one. Ranking in all algorithms is based on spatiotemporal position.

Estimation means finding the missing value from the specified sensor at specified time based on the values of other samples available and their rankings.

III. ALGORITHMS PRESENTATION

A. Naive K Nearest Neighbor

Rank: Euclidian distance with X and Y coordinates and timestamp (1). The difference between the scale of X,Y values and timestamp values makes this method unreliable.

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (t_2 - t_1)^2} \tag{1}$$

Estimate: Arithmetic mean of K closest neighbors values. The setback of this approach is that the most relevant and the least relevant samples have the same weight.

B. Separate weighted average

Rank: Dividing samples into two groups comparing to the missing sample:

- All samples from the same sensor (time neighbors)
- All samples with the same timestamps (distance neighbors)

Estimate:

- Weighted average of all time neighbors
- Weighted average of all distance neighbors
- Arithmetic mean of two estimations

Three aforementioned estimation approaches are tested separately to evaluate the performance of distance and time neighbors in estimation.

C. Combined weighted average

Rank: Product of distance and time difference from missing sample, normalized to range [0.1,1.1] (2).

$$X' = a + \frac{(X - X_{min})(b - a)}{X_{max} - X_{min}} \tag{2}$$

Estimate: Weighted (3) average of K lowest ranked samples.

$$\frac{\frac{1}{d*t}}{\sum_{i=0}^{k} \frac{1}{d_i * t_i}} \tag{3}$$

D. Weighted average and linear regression

Rank: Dividing samples into three groups comparing to the missing sample:

- All samples from the same sensor (time neighbors)
- All samples with the same timestamps (distance neighbors)
- The rest

Estimate:

1) K1[1] time neighbors used in estimation by linear regression
2) K2 distance neighbors are used to find weighted average. If the amount of available distance neighbors is less than K2, using the rest group more distance neighbors are generating using 1).
3) Weighted average of two estimation depending on K1 and K2.

IV. Testing

Sensor data for testing was obtained by using The Consolidated Model of Fire and Smoke Transport, CFAST, created by Fire Research Division at NIST [3].

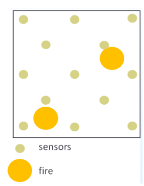


Fig. 2. Room plan used in simulation

Room setting:

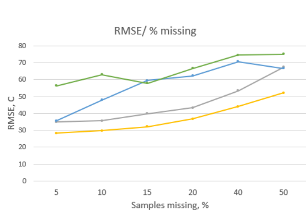
- Single room 20x20 meters
- 2 fires
- 13 sensors
- Sampling interval 5-30 seconds

Tests are performed with changing parameters:

- Number of sensors: 5-13, default is 9
- Sampling interval: 5-30 sec, default is 15 sec
- Samples per package per sensor: 2-10, default is 5
- Percentage of missing samples: 5%-50%, default is 25%
- Percentage of top-ranked samples used in estimation (k): 50%-100%, default is 70%

The error is calculated using Root Mean Square Error (RMSE)(4)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(X_i' - X_i)^2}{N}} \tag{4}$$



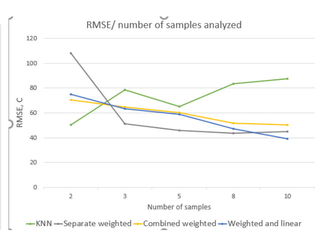
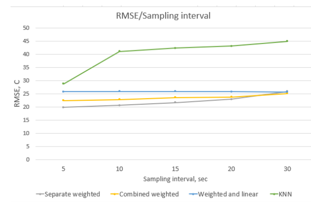


Figure 2. RMSE measured against changing percent of missing samples

Figure 3. RMSE measured against changing number of samples per package per sensor

[1]K1 and K2 are the numbers of neighbors used in estimation, dynamic parameters in testing



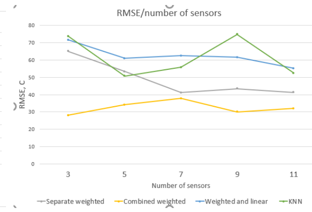


Figure 4. RMSE measured against changing sampling interval

Figure 5. RMSE measured against changing number of sensors in the room

Results indicate that different algorithm perform better in different scenarios. For example, estimating by only samples from the same sensor by weighted average can give the lowest error among all algorithms in particular settings. When the settings change, error increases dramatically. On the contrary, the D*T approach, while still producing considerable error, performs well in changing settings, showing little to no dependence from the amount of available samples, amount of missing samples, etc. Combining distance and time difference into one correlation function with more accurate algorithms could be a potential solution to this problem.

V. Conclusions and Future Work

Missing sensor data samples due to network connection loss, network congestion or sensor malfunction could slow down the analysis of data in smart fire fighting. Estimating those missing samples could make important decisions during emergency more informed. We tested 4 algorithms that use different approaches to solve this problem. Although, more research needs to be done to make the algorithms more accurate, it is clear that the problem should be approached by combining distance and time difference in weighting and ranking the samples. In the future research, more sophisticated design patterns for the estimation could be found that allow for more diverse data.

Acknowledgments

I would like to thank Professor Nalini Venkatasubramanian for hosting me through the IoT-SITY Program. I am also grateful to Dr. Yusuf Sarwar, Georgios Bouloukakis for giving valuable advice throughout the project and Kyle Benson along with Guoxi Wang for helping with building a SCALE Box. I would also like to thank Professor Sharnnia Artis, Professor Nalini Venkatasubramanian and Martha Osegueda for organizing and facilitating the REU program. I would also like to acknowledge NSF for sponsoring and funding the REU program.

References

[1] Pan L, Li J (2010) K-nearest neighbor based missing data estimation algorithm in wireless sensor networks. Science Gate:115-122

[2] Anthony P. Hamins et al. (2015) Research Roadmap for Smart Firefighting. NIST Special Publication 1191.

[3] National Institute of Standards and Technology. CFAST, Fire Growth and Smoke Transport Modeling [Computer software]. Retrieved from https://pages.nist.gov/cfast/index.html