Data Fusion in Wireless Sensor Networks using Fuzzy Set Theory

Ali Berrached and Andre de Korvin
Department of Computer and Mathematical Sciences
University of Houston-Downtown
Houston, Texas 77002 USA
berracheda@uhd.edu dekorvina@uhd.edu

***Abstract*—Wireless Sensor Networks (WSNs) are collections of sensor nodes deployed in a geographical area with the purpose of monitoring the environment in which they are deployed and detect events of interest. Sensor nodes are tiny devices with limited battery power and communication range. Data fusion is the process of combining raw data from the various sensor nodes to obtain information of greater quality and make accurate decisions about the events of interest. Given that energy is the main constrain in WSNs, data fusion can also be used to reduce the volume of data transmitted over the network, thereby extending the network's lifetime. In this paper, we propose a data fusion framework that uses fuzzy set theory to aggregate data from multiple sensors at the cluster level. The algorithm is capable of handling the inherent inaccuracy and conflicts in environmental data readings.**

Keywords-wireless sensor networks; data fusion; data aggregation.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are collections of large numbers of sensor nodes capable of collecting, relaying, and processing sensor readings from the physical world. They have a wide range of applications in both military and civilian environments ranging from natural habitat monitoring to enemy detection and tracking in the battlefield [1]. In most cases, power sources in the sensor nodes are not rechargeable- they are battery based and the nodes are deployed in remote and/or hostile environments. Since a sensor network is usually expected to operate for several months without recharging, energy conservation is an important design objective to prolong the network's lifetime [2]. At the same time, the network's ability to collect and communicate the data of interest on a timely fashion is also a critical objective.

In most applications, sensor nodes are deployed randomly and in large numbers over a target area (e.g., dropped from an airplane). Since data is collected by a large array of densely deployed neighboring nodes, there tends to be a high degree of redundancy and correlation in the data collected. Additionally, due to the harsh conditions in which sensors are often deployed, they tend to be prone to various source of errors such as noise from external sources, hardware noise, sensors inaccuracies and imprecision, and various environmental effects [3]. Data fusion is the process of combining raw data from various sensor nodes. Its main purpose is to obtain information of greater quality and make accurate decisions about the events of interest based on the data collected from the various sensors [4][7][8]. Our proposed data fusion framework uses Fuzzy set theory [6] to handle the inaccuracies that are inherent in sensor data readings and for combining conflicting information from various sources. Moreover, by performing data aggregation at the cluster-level, an additional benefit is the reduction of the amount of raw data transmitted over the network, which prolongs the network lifetime. A secondary benefit is to prevent flooding the Base Station with raw data.

In the next section, we lay out our data fusion framework and its underlying assumptions. In Section 3, we present a simple cluster-level data fusion algorithm that assumes accurate sensor readings. In Section 4, we present a base station-level aggregation algorithm. In Section 5, we present our cluster-level fuzzy fusion algorithm, and in Section 6, we present our concluding remarks and future extensions.

II. DATA FUSION FRAMEWORK

We assume a network of sensors deployed randomly and in large numbers in a bounded area where events of interest are expected to occur. Sensors are assumed to be stationary (i.e., not mobile). They are capable of self-organizing into clusters with one node serving as a cluster head for each cluster. The clustering algorithm partitions the network into groups of nodes each covering a subset of the whole coverage region. All nodes in a cluster send their data readings to their cluster head which aggregates and forwards its

 ISBN: 978-1-61208-296-7

decisions to the base station. The base station collects information coming from all cluster heads and produces the final decisions. Figure 1 depicts a diagram of a cluster based wireless sensor network.

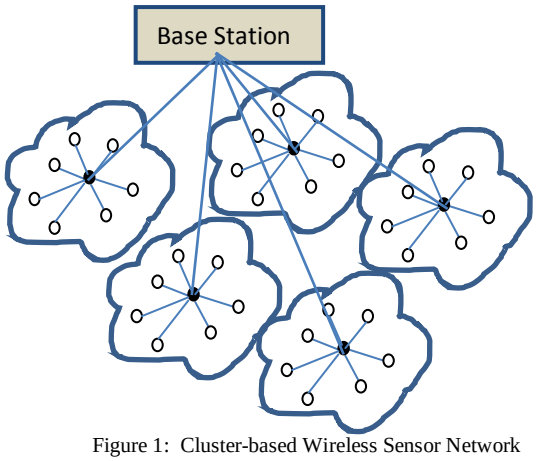


Figure 1: Cluster-based Wireless Sensor Network

There are several clustering algorithms proposed for WSNs [5]. We assume that each cluster consists of N sensor nodes, with N possibly different for each cluster, and each node is capable of sensing one feature or attribute of interest such as the surrounding temperature, humidity, light intensity, smoke density, amount of Carbone Monoxide (or any other chemical), etc.

We propose a framework whereby a *data fusion processor* is employed at the cluster-level. The *data fusion processor* serves to aggregate the raw data transmitted by the cluster nodes and generate decisions that are transmitted to the base station.

III. CLUSTER-LEVEL DATA FUSION

In each cluster, we assume one sensor is activated to monitor one of the features of interest. Assuming there are *m* features to be monitored, there will be *m* sensors with the i^{th} sensor observing feature F_i. The *m* readings are to be aggregated by the cluster *data fusion processor* to reach a decision concerning the occurrence of an event of interest (e.g., the intrusion of an enemy or the occurrence of a fire).

We assume each cluster fusion processor is provided with a local decision matrix **D** defined as:

$$\mathbf{D} = [D_1, D_2, \ldots, D_n]$$

where D_k is a vector of feature values ($[f_{1,k}, f_{2,k}, \ldots, f_{m,k}]$) supporting decision d_k.

As an example, let us consider an application where a sensor network is used for the detection of fire in a forest. The set of monitored features may consist of the following:

F_1: temperature
F_2: smoke density
F_3: humidity

The set decisions:

d_1: Fire unlikely
d_2: Fire likely

A decision matrix can be defined as:

$$\begin{bmatrix} \llcorner 70 & 130\urcorner \\ \llcorner 10 & 70\urcorner \\ 70\urcorner & \llcorner 20 \end{bmatrix}$$

where $\llcorner x$ indicates any value smaller than x is replaced with x and $x\urcorner$ indicates any value larger than x is replaced with x. The above example decision matrix indicates that if feature F_1 (temperature) is 70 (degrees Fahrenheit) or less, feature F_2 (smoke density) is 10 or less, and F_3 (humidity) is greater or equal to 70 then decision d_1 is taken (i.e., Fire unlikely); and if F_1 (temperature) is 130 (degrees Fahrenheit) or more, F_2 (smoke density) is 70 or more, and F_3 (humidity) is 20 or less then decision d_2 is taken (i.e., Fire likely).

Given an actual sampling/reading $\mathbf{R} = [r_1, r_2, \ldots, r_m]$ collected by a cluster from its *m* sensors, the cluster *data fusion processor* takes decision d_j so that:

$$\sum_{i=1}^{m}(f_{i,j} - r_i)^2) \leq \sum_{i=1}^{m}(f_{i,k} - r_i)^2 \quad (1)$$

for all k = 1, 2, …n.

Note that $\sum_{i=1}^{m}(f_{i,k} - r_i)^2$ is a measure of how close the actual data collected by the sensors is to the feature values expected to support decision d_k. Conversely, we can define the strength of a decision d_k as the inverse of the Cartesian distance:

$$Strength(d_k) = \frac{1}{\sum_{i=1}^{m}(f_{i,k} - r_i)^2)} \quad (2)$$

The larger the distance $\sum_{i=1}^{m}(f_{i,k} - r_i)^2$ of the data reading **R** to the feature values expected to supported decision d_k, the weaker the decision.

For the example decision matrix above, if the data reading R = [140, 50, 30] then:

Strength(d_1) = 1/($(70-140)^2$ +$(10-50)^2$ + $(70 - 30)^2$) = 1/8100

Strength(d_2) = 1/($(130-130)^2$ +$(70-50)^2$ + $(20-30)^2$) = 1/500

Decision d_2 (*Fire like*ly) has more strength which, intuitively, is the expected decision for the example reading.

IV. BASE STATION-LEVEL DECISION FUSION

Assuming there are S clusters (where S can vary from cluster to cluster), the base station receives S decisions d_{j_1}, d_{j_2}, …d_{j_s}, one from each cluster head. If the clusters are disjoint, then each cluster decision is representative of the data collected from the area covered by its sensors. On the other hand, if the clusters have overlapping coverage then a number of aggregation algorithms can be used to select a decision from the S decisions received. The simplest method would be to use a simple voting scheme, where the selected decision is the one generated by the most clusters. A better approach would be to use the strength of each decision in the selection process, where the strength of a decision is defined in equation (2). The disadvantage of this approach is that each cluster head needs to transmit, in addition to its decision, the decision strength of its decision. It can, however, lead to more accurate decisions. The additional information can, for example, be used to weed out weak decisions, by considering only decisions that have decision strength above a certain threshold. This can reduce any bias caused by clusters that don't have enough information to make a strong decision (because they are far from the event of interest, for example). From the remaining decisions, we can select the decision that has the majority of votes, or the one with the largest average strength.

V. FUZZY DATA FUSION

The data fusion process described above assumes exactly one sensor is deployed to monitor each feature and sensor readings are accurate and precise. Most often, however, environmental data tends to be vague and noisy, and sensors tend to be prone to errors and malfunction. To overcome these limitations, multiple sensors can be activated to monitor each feature of interest. Fuzzy set theory allows us to map inaccurate crisp sensor readings into fuzzy values that include a measure of confidence or belief of the accuracy of the readings. It also allows us to aggregate conflicting data readings. We first give a brief overview of fuzzy sets.

A. *Overview of Fuzzy Sets*

A fuzzy set A on space X is defined by its membership function:

$$A: X \rightarrow [0, 1]$$

The membership function is a generalization of the characteristic function of a crisp (i.e., ordinary) set. For each $x \in X$, A(x) denotes the degree to which element x is a member of fuzzy set A. For a crisp set, of course,

$$A(x) = \begin{cases} 1 \; iff \; x \in A \\ 0 \; Otherwise \end{cases}$$

For fuzzy sets, $0 \leq A(x) \leq 1$. Those x's for which $A(x) > 0$ constitute the support of fuzzy set A. For notational convenience, we do not distinguish between the membership function and the fuzzy set itself. In effect, the membership function is the fuzzy set. When the domain $X = \{x_1, x_2, \ldots, x_n\}$ is finite, we represent fuzzy set A by the notation:

$$A = \sum_{i=1}^{n} \mu_i / x_i$$

where $\mu_i = A(x_i)$ denotes the degree to which x_i belongs to *A* or the confidence of the belief that x_i belongs to *A*

B. *Data Fuzzification*

A stated above, we assume within each cluster, each feature is observed by an array of *s* sensors. Therefore, for each of *m* features F_i, *s* readings are collected by each cluster. The process of data fuzzification begins by partitioning the domain of each feature F_i into a set of *v* intervals, I_1, I_2, …, I_v and mapping each of the *s* readings into the interval that it belongs to. For each interval I_k, we can get the fraction α_{ik} of sensors observing feature F_i whose reading falls in interval I_k.

If we represent each interval I_k with its mid-point P_k, we obtain a function:

$$f_i: P_k \rightarrow \alpha_{ik}, \text{ for all } k = 1, 2, \ldots, v$$

where *v* is the number of sampling intervals and *i* is the i^{th} feature. After normalizing f_i and taking a linear interpolation, we obtain a continuous function $\bar{f}_i$ that peaks at 1. The normalized values α_{ik} for feature F_i

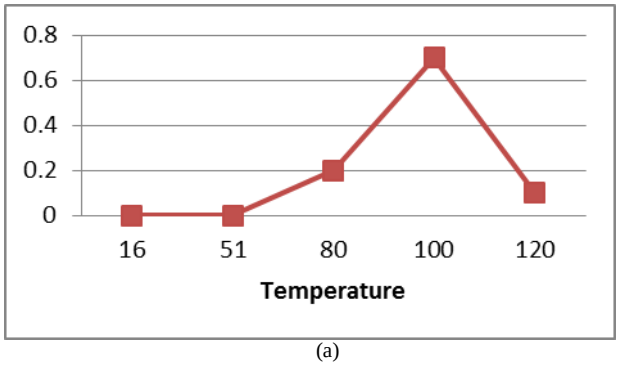
 ISBN: 978-1-61208-296-7

represent the confidence level that the value of feature F_i is P_k.

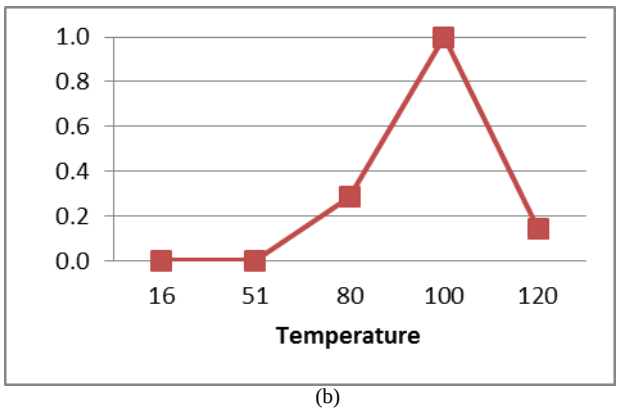
$$\hat{f}_i : x \in X \rightarrow \mu_x, \text{ where } \mu_x \in [0, 1]$$

As an example, let us assume a given cluster has 10 temperature sensors with the following readings: 97, 95, 78, 99, 102, 98, 97, 82, 119, 96. Let us also assume we have 5 sampling intervals I_1=[0, 32[, I_2=[32,70[, I_3=[70, 90[, I_4=[90, 110[, and I_5=[110, 130]. Alternatively, we can express these temperature intervals in linguist terms such very low, low, medium, high, and very high.

Figure 2(a) shows the fraction of readings for each of the sampling intervals represented with the interval's midpoint and Figure 2(b) shows its normalized counterpart.



(a)



(b)

Figure 2: Example data fuzzification

Based on Figure 2(b), the temperature value deduced from the 10 sensor readings is

$$temperature = 0.28/80 + 1/100 + 0.14/120$$

indicating a high confidence that the temperature is close to 100 and lower confidence for the other two values.

C. Cluster-Level Decision Making

As in the crisp case, each cluster head uses a decision matrix:

$$\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_n]$$

where $\mathbf{D}_l$ is a vector $[f_{1,l}, f_{2,l}, \ldots, f_{m,l}]$ of the ideal feature values supporting decision d_l. Ideally, if the feature values computed by a cluster head (based on its senor readings) are exactly equal to vector D_l, then the cluster head (i.e., the data fusion processor) should take decision d_l. However, in general the computed value will not match exactly any of the vectors D_l. The objective of the decision making process is to select the decision that matches best the computed features values.

Let $\mathbf{R} = [\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_m]$ be a set of fuzzy membership functions computed by a cluster head based on its sensor readings, where $\hat{r}_i$ is the membership function for feature F_i. Note that each $\hat{r}_i$ is a fuzzy membership function computed using the procedure described in the previous section. Using the notation of the previous section:

$$\hat{r}_i = \sum_{k=1}^{n} \mu_{ik} / x_{ik}$$

where $\mu_{ik} \in [0, 1]$ for all i and k.

We define the strength of decision d_l as:

$$\text{Strength}(d_l) = \sum_{i=1}^{m} min\left(\sum_{i=1}^{k} max\left(\mu_{ik}\, e^{-(x_{ik}-f_{i,l})^2}\right)\right) \quad (3)$$

Noting that the *min* function produces the weakest link among a set of series links and the *max* function generates the strongest link among a set of parallel links, we use the *max* function to weed out readings that either have low confidence level or large distance to the ideal value for each feature, then we use the *min* function to represent the strength of a decision with the strength of its weakest feature reading. Other aggregate functions such as the mean can also be used instead of the *min* and the *max* functions.

As an example, let us assume the decision matrix below with 2 features F_1 and F_2 and 3 decisions d_1, d_2, and d_3:

$$\begin{matrix} d1 & d2 & d3 \end{matrix}$$
$$\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

 ISBN: 978-1-61208-296-7

Assume at a given cluster, we get the following feature membership values based on our sensor readings and using the fuzzification procedure of the previous section:

$$\hat{r}_1 = 0.8/1 + 0.2/3$$
$$\hat{r}_2 = 0.4/3 + 0.6/4$$

Intuitively, the readings support decisions d_1 and d_2, while decision d_3 should be the weakest. Applying equation (3), we get the strength of each decision:

Strength(d_1) = min[(max($0.8\, e^{-(1-1)^2}$, $0.2\, e^{-(3-1)^2}$), max($0.4\, e^{-(3-2)^2}$, $0.6\, e^{-(4-2)^2}$)]

Strength(d_1) = min(max(0.8, 0.0036), max(0.268 , 0.011)) = 0.268

Similarly, we compute
Strength(d_2) = min(max($0.8\, e^{-(1-3)^2}$, $0.2\, e^{-(3-3)^2}$), max($0.4\, e^{-(3-4)^2}$, $0.6\, e^{-(4-4)^2}$))

Therefore, Strength(d_2) = min(0.2 , 0.6) = 0.2

Strength(d_3) = min(max($0.8\, e^{-(1-5)^2}$, $0.2\, e^{-(3-5)^2}$), max($0.4\, e^{-(3-6)^2}$, $0.6\, e^{-(4-6)^2}$))

Therefore, Strength(d_3)= min(0.00366, 0.011) = 0.00366.

Based on the above results, decision d_3 is the weakest, which intuitively is what we expected, while decision d_1 and d_2 are stronger, with d_1 slightly stronger than d_2.

VI. CONCLUSION AND FUTURE WORK

Data fusion in wireless sensor networks is critical to reduce raw data transmission across the network, which would in turn increase the network's lifetime and prevent flooding the base station. In this paper, we presented a cluster-level data fusion algorithm based on fuzzy set theory that's capable of handling inaccurate and conflicting sensor readings. We plan to extend this algorithm to aggregate sensor readings over time so that sensor nodes update their cluster heads only when significant changes occur in the sensed feature readings, and in turn, cluster heads update the base station only when a significant change occurs in the decision taken or it confidence level. We also plan to further investigate decision fusion among partially overlapping clusters.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cyirci, "Wireless sensor networks: A survey," Computer Networks 38(4), 2002, pp. 393–422.

[2] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, "Efficient energy management in sensor networks," Ad Hoc and Sensor Networks, 2005, pp. 71-90

[3] E. Elnahrawy and N. Badri, "Cleaning and querying noisy sensors," Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, 2003, pp. 78-87.

[4] R. C. Lou, C. Yih, and K. L. Su, "Multisensor fusion and integration: Approaches, applications, and future research directions," IEEE Sensors J. 2, 2 (April 2002), pp. 107–119.

[5] J. C. Maxwell, "A Treatise on Electricity and Magnetism," 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp. 68–73.

[6] W. Pedrycz and F. Gomide, "An Introduction to Fuzzy Sets Analysis and Design," MIT Press, Cambridge, Massachusetts, 1998.

[7] L. WALD, "Some terms of reference in data fusion," IEEE Trans. Geosci. Remote Sens., 13, 3 (May 1999), pp. 1190–1193.

[8] N. Xiong and P. Svensson "Multi-sensor management for information fusion: issues and approaches," Information Fusion, 2002, vol. 3, pp. 163–186.

 ISBN: 978-1-61208-296-7